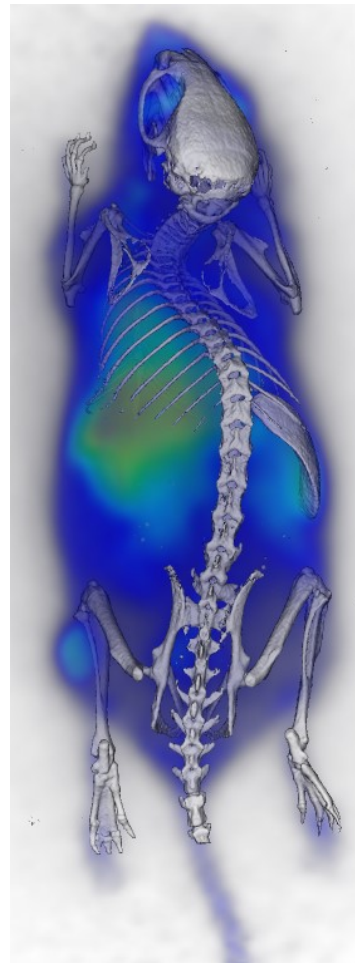
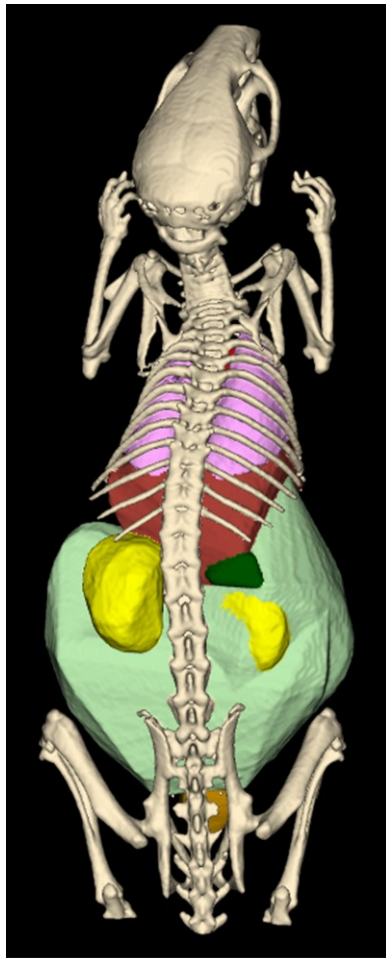


IMALYTICS

Preclinical 2.1



Version 2.1.8.5
May 2019

User's Manual

Table of contents

1. INTRODUCTION.....	4
PURPOSE OF THE SOFTWARE - <i>IMALYTICS PRECLINICAL</i>	4
POSSIBLE APPLICATION NOTES	5
<i>Cancer research</i>	5
<i>Blood system analysis</i>	5
<i>Pharmakokinetic</i>	5
<i>Gastroenterology</i>	5
<i>Translational medicine</i>	5
<i>Other applications</i>	5
TECHNICAL REQUIREMENTS	5
2. INSTALLATION	6
INSTALLATION OF THE SOFTWARE.....	6
INSTALLATION OF GRAPHIC CARD DRIVER	7
3. GETTING TO KNOW THE PROGRAM- FIRST STEPS	8
STARTING THE PROGRAM AND DISPLAY YOUR IMAGE DATA	8
VIDEO TUTORIALS	8
OVERVIEW-MAIN WINDOW	8
TOOL BAR	9
MOUSE ACTION MODES (LEFT BUTTON).....	10
<i>Mouse action modes (selected by buttons)</i>	10
<i>Other mouse operations</i>	11
CONTEXT MENU.....	11
4. THE MENU BAR	13
FILE.....	13
EDIT	15
RESAMPLE	16
VIEW.....	18
OPTIONS.....	19
CLASSES.....	20
STATISTICS.....	21
SEGMENTATION	25
MATH	27
MARKERS.....	28
TIME	29
CHANNELS.....	30
SCRIBBLES	30
BATCH	32
RENDERING	32
UNMIXING	33
FUSION.....	33
BIOLUMINESCENCE.....	35
RELAXOMETRY	35
VESSELS	36
3D PRINTING	37
MACROS.....	38
BONE	39
HELP	40
5. GPU PROCESSING.....	41

6. FREQUENTLY ASKED QUESTIONS	41
<i>HOW CAN I SAVE A FLY-THROUGH MOVIE OF MY IMAGE?</i>	41
7. TROUBLESHOOTING.....	41
<i>HOW TO PROVIDE FEEDBACK?</i>	41
8. REFERENCES.....	42

Contact

Dr. Felix Gremse

Gremse-IT GmbH
Pauwelsstr. 20
D-52074 Aachen
gremse@gmx.net

+49 241 80 89761

1. Introduction

Purpose of the software - *IMALYTICS Preclinical*

The purpose of this software is to visualize and analyze multimodal preclinical and medical image data sets using interactive segmentation operations. This program was developed by Felix Gremse at the Institute for Experimental Molecular Imaging (ExMI), RWTH Aachen University Clinic, in cooperation with Philips Research Aachen, and was commercialized by Gremse-IT GmbH, a spin out of ExMI.

The software is described here [1], [2], please cite one of these references when using the software:

[1] F. Gremse, et al., “Imalytics Preclinical: Interactive Analysis of Biomedical Volume Data,” *Theranostics*, vol. 6, no. 3, pp. 328–341, 2016.

[2] F. Gremse et al., “Hybrid μ CT-FMT imaging and image analysis,” *J Vis Exp*, no. 100, p. e52770, Jun. 2015.

The software is preferably used for fast and accurate analysis of preclinical μ CT or clinical CT data sets. But it is also suitable for the analysis of data acquired by ultrasound (US) [3]–[5], positron emission tomography (PET) [6], single-photon emission computed tomography (SPECT), bioluminescence, and magnetic resonance (MR) imaging [7], [8]. The software is recommended for interpretation of multimodal imaging studies such as PET-CT, FMT-CT, SPECT-CT, MRI-CT, PET-SPECT, and more [2], [4], [9]–[25]. The software is mainly used for preclinical data sets but it is also increasingly used for clinical data sets.

The program allows processing and visualization of three data sets:

- an underlay (e.g. CT in case of μ CT-FMT imaging)
- a segmentation map (e.g. organ segmentation), and
- an overlay (e.g. three-dimensional fluorescence distribution)

The segmentation map is a data set of the same size as the underlay. Each voxel of the segmentation map is assigned to exactly one class, e.g. “Bladder”, “Liver”, or “unclassified” which is the default. The segmentation is useful for quantification, because parameters such as volumes and mean intensities can be computed.

Most often, 3D data sets are processed, but 2D, 4D, and even 5D data sets can also be loaded and analyzed. Typically, the fourth dimension is the time, as in for longitudinal PET scans [6]. The fourth dimension can also represent different echo times from MRI relaxometry measurements [8]. For 4D data sets, the segmentation remains a 3D data set, however. The fifth dimension usually represents multiple channels, e.g. from dual energy CT or from scanning with multiple wavelengths in the FMT, however, the 5D functionality requires a special license.

Possible application notes and references

Cancer research

- ✚ Tumor detection and analysis [18], [26]–[28]
- ✚ Liver and lung metastases [29]
- ✚ Tumor, metastases volumes and blood vessel density [30]–[33]

Blood system analysis

- ✚ Analysis of blood vessels [1], [34]
- ✚ Angiogenesis [35]–[39]
- ✚ Stenosis measurements [40]
- ✚ Atherosclerotic lesions [40]–[45]

Pharmakokinetic

- ✚ Pharmacokinetic and biodistribution assessment of novel drugs, therapeutics, drug carrier systems [18], [19], [46], [47]
- ✚ Assessment of antibody biodistribution in vivo, immunotherapy monitoring [17], [18], [48]

Gastroenterology

- ✚ Fat segmentation, subcutaneous and visceral fat ratios, gastroenterological issues [12], [49], [50]
- ✚ Hepatic vascular regeneration [33]
- ✚ Organ volumes [1], [32]

Translational medicine

- ✚ Calcifications [51]–[53]
- ✚ Dentistry, craniofacial surgery, skull/bone structures [54]–[57]

Other applications

- ✚ Ballistic experiments [58]
- ✚ Multimodal image analysis [59]
- ✚ Organ and tumor segmentation [60]–[62]

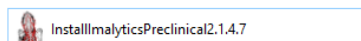
Technical Requirements

This program is available for Windows operating systems. It runs at permanently installed PCs or laptop computers. Furthermore, it requires an NVIDIA GPU graphic card with Cuda compute capability 3.0 or higher (<https://developer.nvidia.com/cuda-gpus>). Without a GPU, the program will not start. The program requires a license. Without a license, or if the license is expired, it will not start either.

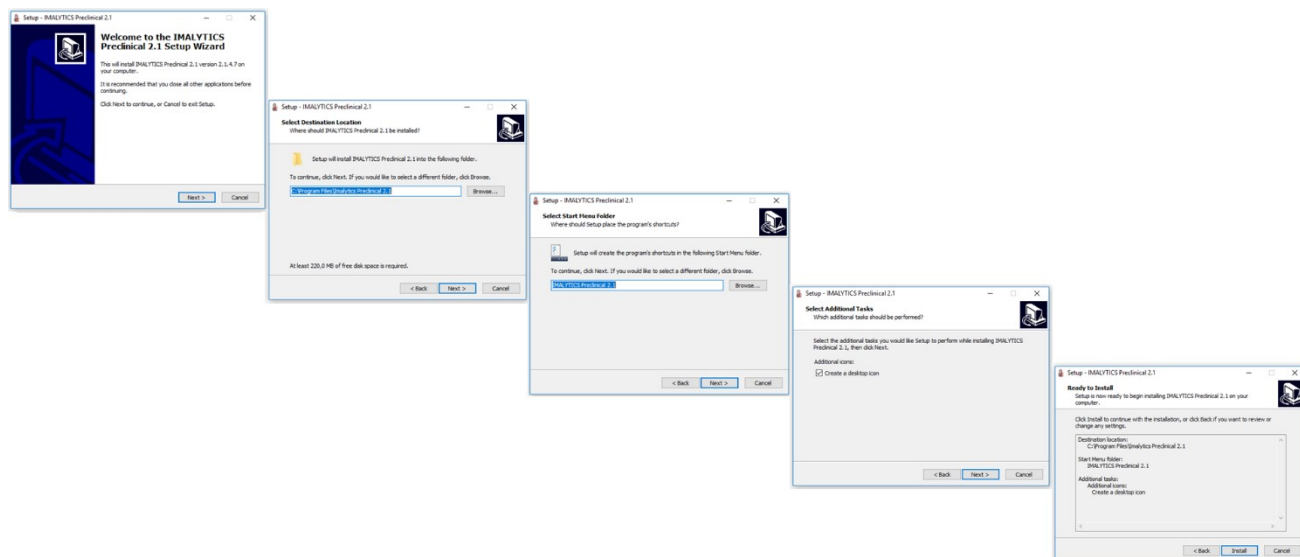
2. Installation

Installation of the software

Run the install file of the software:



Follow the installation steps, the default settings should be sufficient in most cases.



Welcome to the IMALYTICS Preclinical 2.1 Setup Wizard. To start the installation, press *Next* >

Select Destination Location. Select a folder where you want to install the software. Make sure that enough free space (at least 220,0 MB) is available there. To continue the installation, press *Next* >

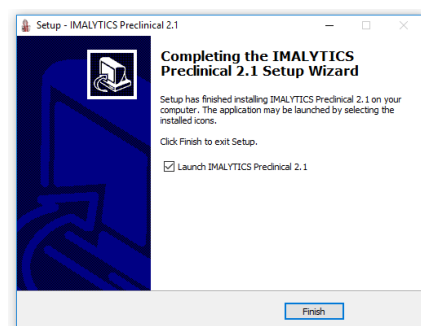
Select Start menu Folder. Select the folder where you want to create the program's shortcut. To continue the installation, press *Next* >

Select Additional Tasks. Decide if you want to create a desktop icon → if yes: activate the tick box by clicking. To continue the installation, press *Next* >

Ready to install. Check your installation setups. If they are correct, start the installation process by clicking *Install*

Wait until the installation is finished (depiction of a green progress bar).

The software will inform you when the installation was successfully completed:

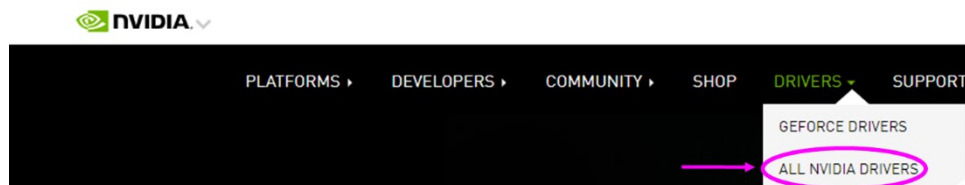


Completing the IMALYTICS Preclinical 2.1 Setup Wizard. If you want to directly launch the software, activate the tick box by clicking. To exit the setup, press *Finish*

Installation of graphic card driver

The software requires an NVIDIA GPU graphic card and a respective driver which is freely available at NVIDIA (<http://www.nvidia.com/page/home.html>). Upon starting, Imalytics Preclinical may complain and require to install a new driver.

- At first, ascertain what graphics card is included in your computer. There are two ways to check:
 - 1) Open the control panel application > Hardware and Sound > Device manager > Graphic card > click right mouse button > Settings
 - 2) Press windows (■) + R → Enter **dxdiag** → wait for one moment > Display (1) → Name: Graphic card, for example **NVIDIA Quadro P5000**
- Then, please open the NVIDIA website (<http://www.nvidia.com/page/home.html>)
- Open drivers → ALL NVIDIA DRIVERS



- Start Manual Driver Search

NVIDIA Driver Downloads

Option 1: Manually find drivers for my NVIDIA products. [Help](#)

Product Type:	Quadro
Product Series:	Quadro Series
Product:	Quadro P5000
Operating System:	Windows 10 64-bit
Download Type:	Optimal Driver for Enterprise (ODE) ?
Language:	English (US)

[SEARCH](#)

Option 2: Automatically find drivers for my NVIDIA products. [Learn More](#)

[GRAPHICS DRIVERS](#)

- Choose Option 1 and enter your graphic card details.
- Press **SEARCH**.
- Download the driver by pressing the button **DOWNLOAD**.
- Follow the download instructions and install the driver.

3. Getting to know the program- First steps

Starting the program and display your image data

After successful installation of the software, open the program by double clicking on an appropriate image file or open the software from the start icon. You can also create a desktop icon (right-click on Imalytics Preclinical and select “send to desktop (shortcut)”. If you open the program directly, you can open your image data by pressing **File > Load underlay (Ctrl + O)** and selecting an image file.

Video Tutorials

Under **Menu > Help > Show video tutorials**, you can play several movies showing how to use this software. The movies have built-in subtitles in English, Chinese, German and French which can be shown with video player such as VLC.

Overview-Main Window

The main window (Figure 2) consists of a menu bar, a tool bar, several panels to show slice and 3D views, and a status bar. Axial, coronal, and sagittal slice views and the 3D view can be toggled independently. The viewing center (a point in 3D) is shown in all windows as red marker. The active point is shown in green, can be set with a right click, and is used by many segmentation operations as seed point.

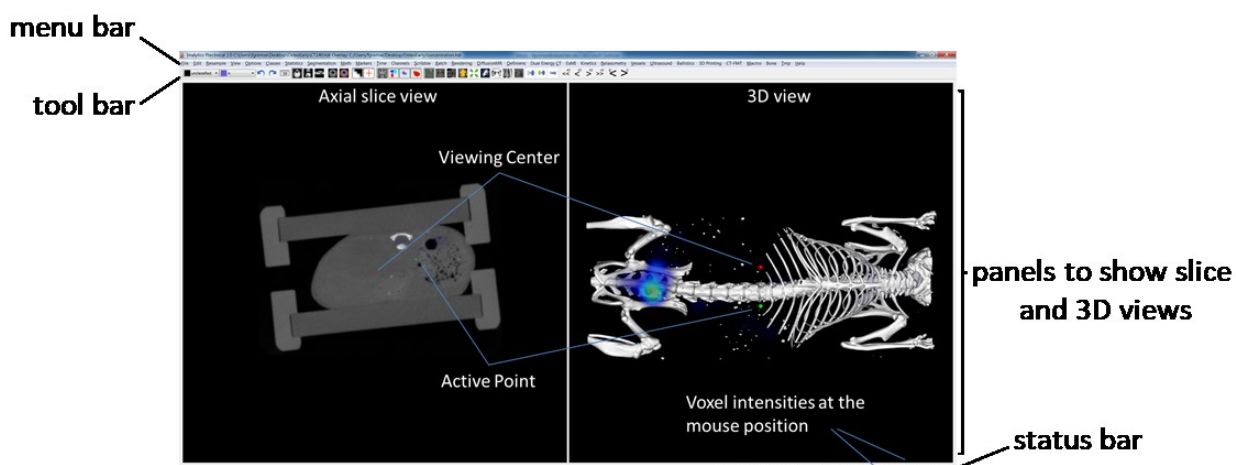


Figure 1: Main window consisting of menu bar, tool bar, panels to show slice and 3D views, and status bar. The menu bar offers different operation modes. The tool bar is mainly used for viewing and segmentation operations. The majority of the space is taken by the panels to show slice and 3D views, here only the axial view and the 3D view is shown. The status bar provides information about the voxel intensities.

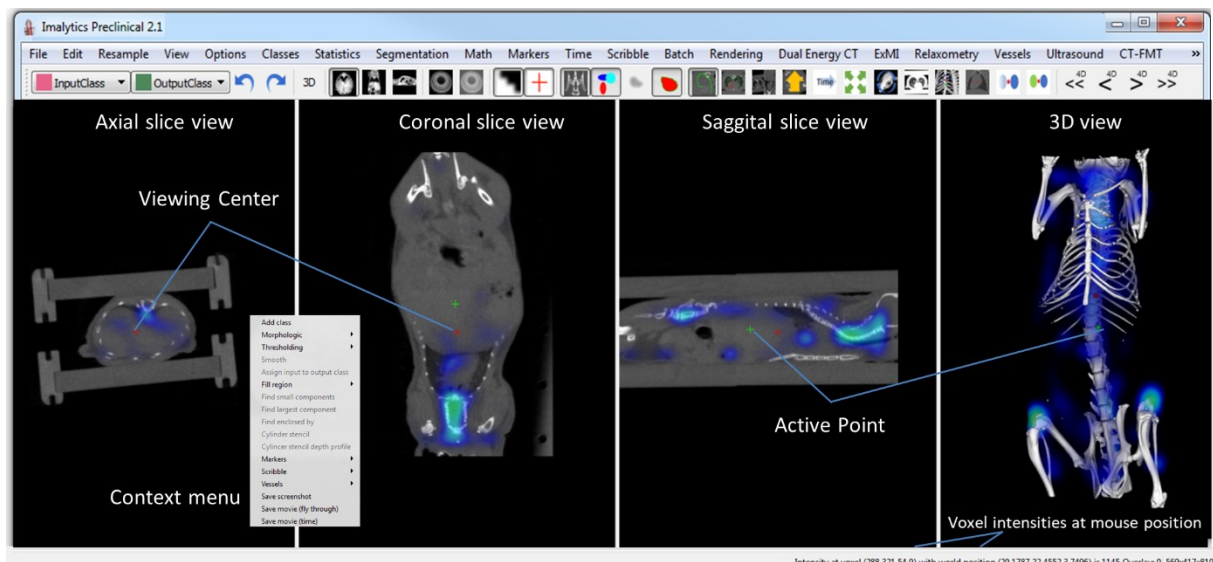


Figure 2: Main window in detail. Axial, coronal, sagittal, and 3D views can be shown next to each other. The viewing center and the active point are indicated by red and green markers. The context menu is displayed (after clicking the right mouse button). The status bar in the bottom right corner shows information about the underlay and overlay voxel intensities at the current mouse position and also information about the size of the data set. Anatomical information are acquired from the μ CT. Fluorescence intensity distribution (FMT) is depicted as a color-coded overlay.

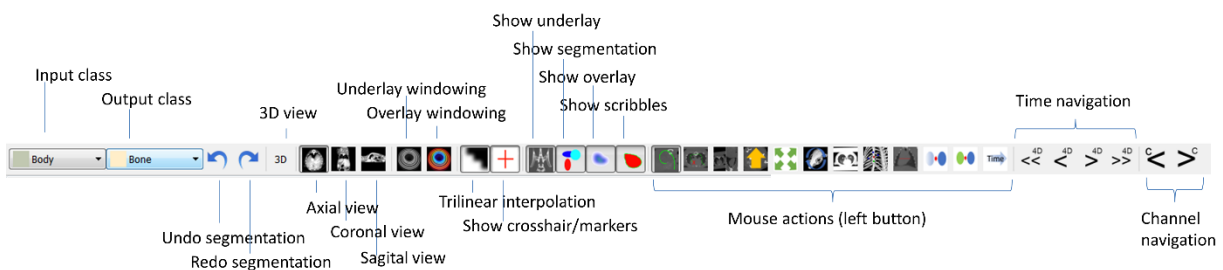


Figure 3: Tool bar. Output and input classes can be selected. Undo and Redo can be performed but these only apply to segmentation operations. Axial, coronal and sagittal slice views and the 3D view can be toggled independently. The windowing dialogs can be opened for underlay and overlay. Trilinear interpolation can be turned on and off. The visualization of crosshairs and markers can be toggled. The rendering of underlay, segmentation, overlay and scribbles can be toggled. Different mouse actions (draw scribbles, delete scribbles, set viewing center, slice through, zoom, rotate view, windowing, set marker, measure distance) can be selected. Navigation is possible through 4D phases and channels if available.

Tool Bar

The tool bar directly appears when you open the software. If the tool bar is disappeared, click on the menu bar with the right mouse button and activate the tick box. The tool bar will immediately appear. Most functions are described in Figure 3.

Mouse action modes (left button)

The mouse can be used for many different operations, e.g. to select the viewing center, or to draw scribbles. Some interactions work in the 2D views and some in 3D but most work in both. The current mouse mode can be selected using the mouse action buttons in the tool bar (Figure 3). Alternatively, they can be selected using special combinations of mouse keys:

LMB = left mouse button

MMB = middle mouse button

Wheel = scrolling-wheel

RMB = right mouse button

and keyboard keys (**SHIFT**, **ALT**, and **CTRL**) as listed below.

Mouse action modes (selected by buttons)



or **SHIFT + LMB: Draw scribbles** (2D only).



or **SHIFT + RMB: Delete scribbles** (2D only). Handy when some wrong scribbles were drawn.



or **CTRL + LMB: Set view center**. Center the view onto the point of the mouse click (2D and 3D). Handy when you want to center for example a special organ, tumor or metastasis.



or **MMB/Wheel: Move/scroll through slices fast** (2D only).



or **SHIFT+Wheel: Move through time**.



or **CTRL + MMB/Wheel: Zoom in and out** (2D and 3D).



or **ALT + LMB: Rotate the view interactively** (2D and 3D).



or **ALT + RMB: Change windowing settings interactively** (2D and 3D).



Create a marker at the mouse click position (2D and 3D).



or **CTRL + RMB: Measure distance interactively** (shown in lower right corner) (2D only).



or **CTRL + SHIFT + LMB: Move a class around** (2D only).



or **CTRL + ALT + LMB**: Copy class to new position (copy gets new name) (2D only).

Other mouse operations

RMB: Sets output class and opens context menu (2D and 3D).

MMB/Wheel: Change the output class (2D and 3D).

ALT + MMB/Wheel: Move through slices one by one (2D and 3D).

SHIFT + MMB/Wheel: Go through time points one by one (2D and 3D).

SHIFT + MMB/Wheel: Go through time fast (2D and 3D).

SHIFT + ALT + LMB: Repeat previous fill region operation (2D and 3D).

Context Menu

The context menu appears when clicking the right mouse button (Figure 4).

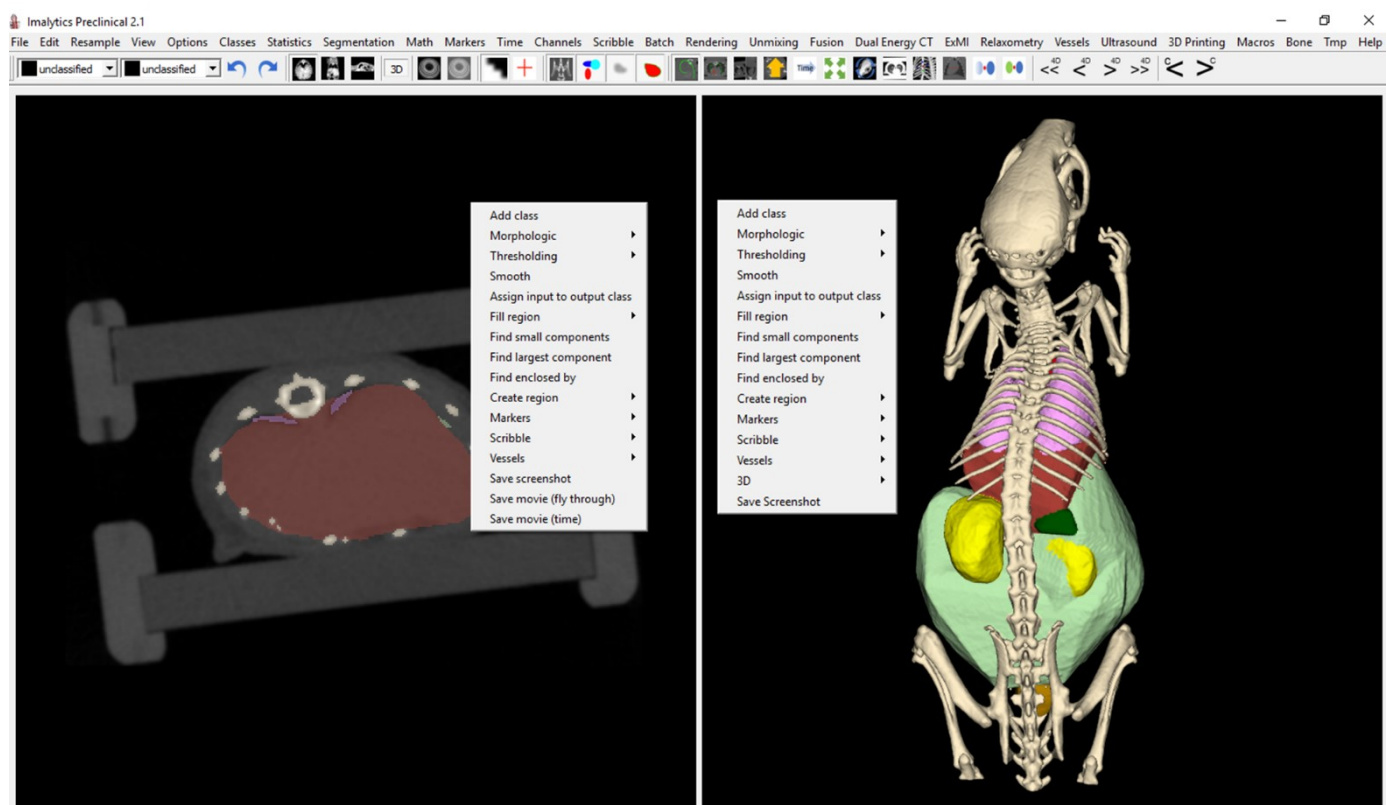


Figure 4. Context menu. It appears when right-clicking which also sets the active point (green crosshair(2D) or sphere (3D)) and the input class. The context menu is populated with many operations which can also be reached through the menu bar, e.g. “Add class” can be called through Menu->Classes->Add class. These operations are explained in detail in the section >Menu bar<. **Save screenshot** saves the panel as .png file. **Save movie (fly through)** creates a .wmv movie flying through the slices. **Save movie (time)** saves a movie going through time (requires 4D data sets). The context menu also appears in 3D with slightly different entries, e.g. to render a rotating movie or so save and load the current camera position.

Context menu > Add Class: Menu bar > Classes > Add Classes

Context menu > Morphologic (Dilate, Erode, Open, Close, Coat): Menu bar > Segmentation

Context menu > Thresholding (Above, Below, Range, Above center): Menu bar > Segmentation

Context menu > Smooth: Menu bar > Segmentation > Smooth input class

Context menu > Assign input to output class: Menu bar > Segmentation

Context menu > Fill region (-unlimited, -volume bounded, -distance bounded, - again, Grow into output class): Menu bar > Segmentation

Context menu > Find small components: Menu bar > Segmentation

Context menu > Find largest component: Menu bar > Segmentation

Context menu > Find enclosed by: Menu bar > Segmentation

Context menu > Cylinder stencil: Menu bar > Segmentation

Context menu > Cylinder stencil depth profile: Menu bar > Segmentation

Context menu > Markers: Menu bar > Markers

Context menu > Scribble: Menu bar > Scribble

Context menu > Vessels: Menu bar > Vessels

Context menu > Save screenshot: Saves the panel as a .png file. Please choose the folder where you want to save the image.

Context menu > Save movie (fly through): Creates a .wmv movie flying through the slices. Choose the distance in the data set [mm] (negative values will invert the direction) and the duration of the movie [s]. When the movie is ready, a window with “*Movie is rendered.*” appears. You have to press *OK*, otherwise the software will not continue working.

Context menu > Save movie (time): Saves a movie going through time (requires 4D data sets). Choose the duration in the data set [s] and the duration of the movie [s]. When the movie is ready, a window with “*Movie is rendered.*” appears. You have to press *OK*, otherwise the software is not working.

- *The context menu also appears in 3D with slightly different entries, e.g. to render a rotating movie or so save and load the current camera position.*

Context menu > 3D **> Render rotation movie (z-axis):** Creates a .wmv movie in 3D flying through the slices. Choose the distance in the data set [mm] (negative values will invert the direction) and the duration of the movie [s]. When the movie is ready, a window with “*Movie is rendered.*” appears. You have to press *OK*, otherwise the software will not continue working.

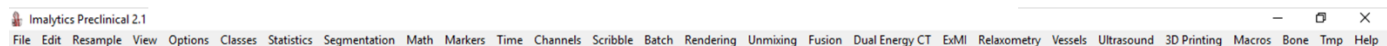
> Render rotation movie (z-axis) for Tridality (5-tiled): Renders a movie that plays in Tridality 3D screens

> Anti aliasing: Toggle anti-aliasing for faster frame rendering

- > **Measure frame rate:** Measures the frame rate of the current view.
- > **Reset Camera position:** Resets the 3D camera position.
- > **Save Camera Position:** Saves the current camera position as a CAM-Datei (.cam).
- > **Load Camera Position:** Loads a camera parameters from a .cam file.
- > **Save Depth map:** Saves the depth map from the current view as image.

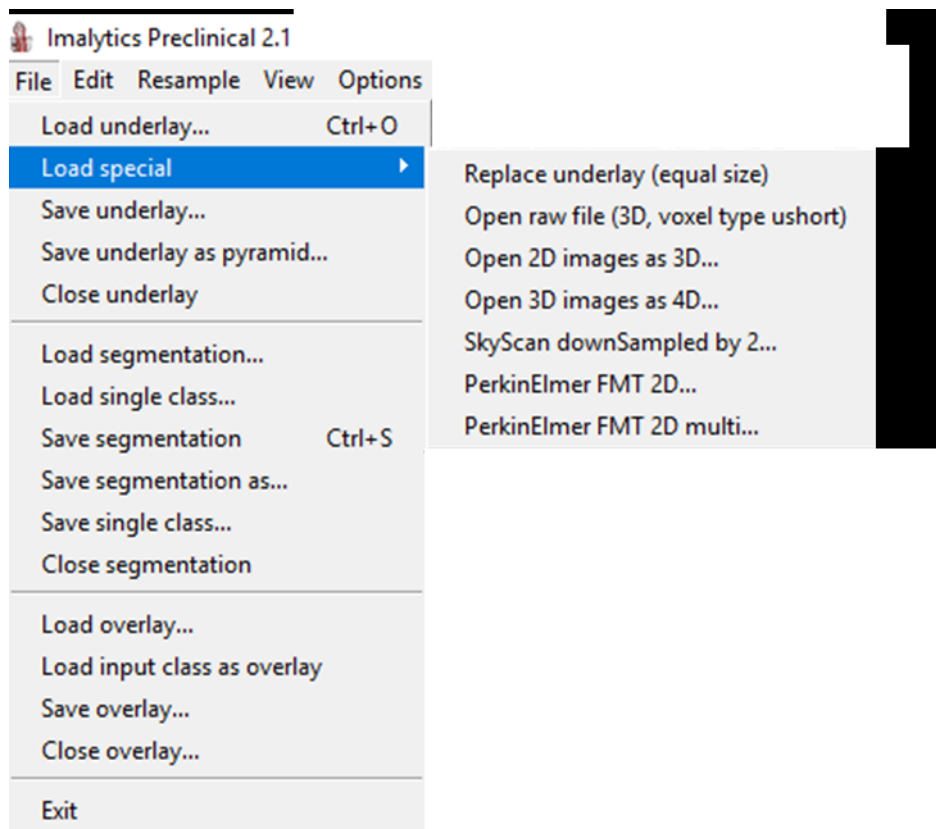
4. The menu bar

The menu bar is split into several sub menus that are described in detail here:



File

The File menu allows you to open and save an underlay file, load/save a segmentation map and load an overlay. Supported file formats are listed in Table 1. The File menu contains the following menu items:



File > Load underlay... (Ctrl + O): Loads an underlay from a user selected file. Supported file formats are shown in table 1.

File > Load special: Allows reading special file formats, to load a set of 2D images as one 3D image, or to load a set of 3D images as one 4D image. Also, loading of raw files is possible if the user can provide the correct dimensions.

File > Create underlay...: Creates or loads an underlay. There are several options: Sphere, cube, cylinder, Menger sponge, Shepp-Logan phantom, CT-FMT example, fusion example...

File > Save underlay...: Saves the underlay as file. Different file format are supported. Some file formats may not support 3D files, resulting in an error message. When selecting PNG, 3D files are stored as multiple files (only for 8bit PNGs).

File > Save underlay as pyramid...: Saves the underlay as pyramid in the nifti file format, i.e. multiple images at decreasing resolutions). The user selects a folder where to save these files. The underlay is repeatedly down-sampled by 2x2x2 binning until the X-dimension is 10 or less.

File > Close underlay: Closes the underlay.

File > Load segmentation...: Loads a segmentation from file.

File > Load single class...: Loads only a single class from file. Adds this class to the current segmentation. The user will be asked to overwrite other classes or not. This can be used to merge segmentation files.

File > Save segmentation (Ctrl + S): Saves the segmentation as .seg file.

File > Save segmentation as...: Saves the segmentation under another file name.

File > Save single class...: Saves a single class as .seg file.

File > Close segmentation: Closes the segmentation.

File > Load overlay...: Loads an overlay from file.

File > Add overlay time point. Loads an overlay from file and appends it in time to the currently loaded overlay. The requires matching 3D/4D dimensions and voxel sizes.

File > Add overlay channels. Loads an overlay from file and adds the channels to the current overlay. The requires matching dimensions and voxel sizes.

File > Load input class as overlay...: This converts the binary mask of a class to visualize it using transparent volume rendering. It replaces the current overlay.

File > Save overlay...: Saves the overlay as file.

File > Close overlay: Closes the overlay.

File > Exit: Closes the program.

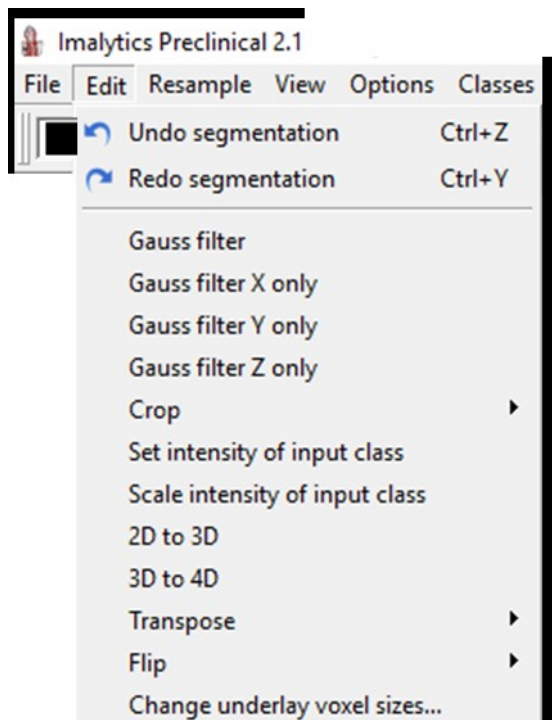
Table 1: Supported file formats for underlay and overlay files. DICOM files can be converted to nifty using Chris Rorden's dcm2nii program (<http://people.cas.sc.edu/rorden/mricron/dcm2nii.html>).

Name	Extension	Reading is supported	Saving is supported
Analyze	.hdr/.img	Yes	Yes
Nifti	.nii	Yes	Yes
MicroPET	.hdr	Yes	No
Vevo770	.vsi	Yes	No
Vevo2100	.bimg,animg,nbimg	Yes	No
CT-Imaging	.mif,mifx	Yes	No
TIFF	.tif	Yes	No
MetImageHeader	.mhd	Yes	Yes
PAR/REC	.par/.rec	Yes (partially)	No

NRRD	.nhdr	Yes	Yes
PNG	.png	Yes	Yes
JPEG	.jpg	Yes	No
Skyscan Recons	.log	Yes	No
VFF files	.vff	Yes	No
VOX files	.vox	Yes	No
Bitmap files	.bmp	Yes	No
DICOM	.dcm	Yes (partially)	Yes (partially)

Edit

The edit menu provides operations to edit the underlay file. It also contains the Undo/Redo operations which apply only to the segmentation, however.



Edit > Undo segmentation (Ctrl + Z): Performs an Undo for the last segmentation operation.

Edit > Redo segmentation (Ctrl + Y): Performs a Redo for the last segmentation operation.

Edit > Gauss filter: Applies a Gauss filter to the underlay. You can provide the standard deviation.

Edit > Gauss filter X only: Applies a Gauss filter to the underlay in X direction only.

Edit > Median filter (3x3x3): Applies a local 3x3x3 median filter to the underlay.

Edit > Crop: Crops the underlay. Different modes are available.

Edit > Set intensity of input class: Sets all voxels of the input class to a user-provided value.

Edit > Scale intensity of input class: Scales all voxels of the input class by a user-provided factor.

Edit > 2D to 3D: Sometimes a 3D data set is stored as a 2D image by concatenating all slices below each other. This operation converts this back into a 3D data set.

Edit > 3D to 4D: Sometimes a 4D data set is stored as a 3D image by concatenating all cubes behind each other (in the z-direction). This operation converts this back into a 4D data set.

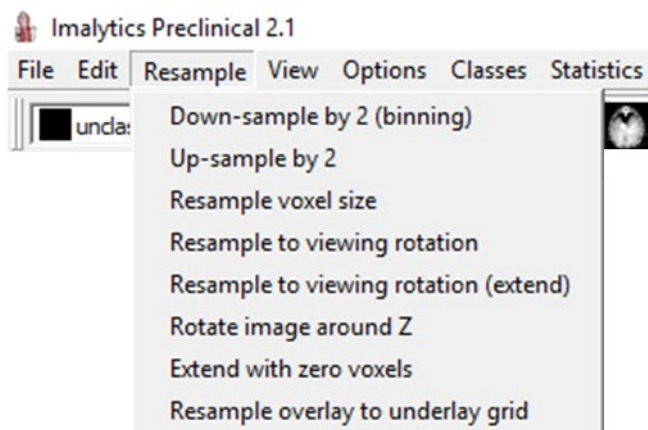
Edit > Transpose > Transpose X and Y, Transpose X and Z, Transpose Y and Z, Transpose X and T, Transpose Y and T, Transpose Z and T: Transposes the data set. Different versions are available, e.g. to swap x and y axes. The time dimension is called T.

Edit > Flip > Flip X, Flip Y, Flip Z: Flips the image around an axis (either x, y or z).

Edit > Change underlay voxel sizes...: Allows the user to enter the voxel sizes of the current underlay.

Resample

The resample menu provides functionality to resample the underlay and the segmentation map. The segmentation map is always transformed using nearest neighbor resampling. The undo segmentation mechanism does not work on these operations because they involve changes on the underlay or overlay.



Resample > Down-sample by 2 (binning): Computes a data set of half the size in each dimension. 8 voxels are averaged (“binned”) into one voxel. This operation reduces the image size and the noise.

Resample > Up-sample by 2: Creates an underlay of higher resolution, i.e. half the voxel size. Trilinear interpolation is used for the underlay.

Resample > Resample voxel size: Performs trilinear interpolation to compute an underlay with a user-provided voxel size.

Resample > Resample to viewing rotation: Performs trilinear interpolation to rotate the underlay according to the current viewing rotation. The viewing rotation can be adjusted interactively with the mouse (ALT+ LMB). The image size remains the same which may lead to some loss of pixels at the borders.

Resample > Resample to viewing rotation (extend): Same as above but automatically increases the image size to keep all voxels.

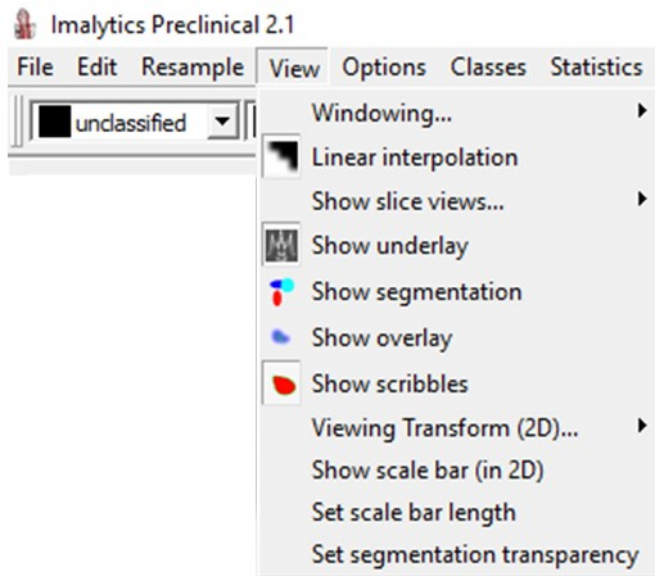
Resample > Rotate image around Z: Rotates the underlay, by a user-provided rotation angle around the Z-axis.

Resample > Extend with zero voxels: Extends the underlay and segmentation with a border of zero voxels. The border width is specified by the user.

Resample > Resample overlay to underlay grid: Performs trilinear interpolation to resample the overlay to the underlay voxel grid.

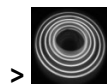
View

The view menu provides functionality to adjust the viewing settings.



View > Windowing ...: The sub menu windowing has items for handling the windowing of underlay and overlay.

- > **Reset underlay windowing (F5):** Resets the underlay windowing to min/max values, based on the current 3D image in case of a 4D underlay.
- > **Reset overlay windowing (F6):** Resets the overlay windowing to min/max values, based on the current 3D image in case of a 4D overlay.
- > **Select underlay color map:** To select a different underlay color map.
- > **Select overlay color map:** To select a different overlay color map.




- > **Underlay windowing...:** To adjust the underlay windowing.



- > **Overlay windowing...:** To adjust the overlay windowing.

- > **Save underlay color map:** Saves an image of the current underlay color map.

- > **Save overlay color map:** Saves an image of the current overlay color map (e.g. )



View > Linear interpolation: To toggle whether linear interpolation or nearest neighbor interpolation should be used.

View > Show slice views...: Allows toggling which slice views are shown (Axial (X,Y) – Coronal (X, Z) – Sagittal (Z,Y)).



View > Show underlay: To show the underlay or not.



View > Show segmentation: To show the segmentation or not.



View > Show overlay: To show the overlay or not.



View > Show scribbles: To show the scribbles or not.

View > Viewing transform (2D)...: This sub menu has items for the 2D viewing transform which consists of viewing center (red cross), rotation and zoom. The viewing transform defines the 2D slice views and can be changed interactively with the mouse (ALT+LMB).

> **Reset 2D viewing transform:** Resets the viewing transform of the slice views to be the identity matrix (no rotation).

> **Reset 2D viewing rotation:** Resets the viewing rotation of the slice views.

> **Set viewing zoom:** Allows the user to set the zoom manually.

> **Set view based on start and end markers:** Requires two markers named “Start” and “End”, e.g. at the start and end of a blood vessel segment. Then the viewing rotation is set to be orthogonal to the connecting line. The viewing position is set to the start marker. This can be used to slice through a blood vessel.

> **Save viewing transform:** Saves the viewing transform to a file.

> **Load viewing transform:** Loads the viewing transform from a file.

View > Show scale bar (in 2D): Shows a scale bar in the slice views.

View > Set scale bar length: Sets the scale bar length [mm].

View > Set segmentation transparency: To set the segmentation transparency for slice views between (0 and 1) (in 2D only).

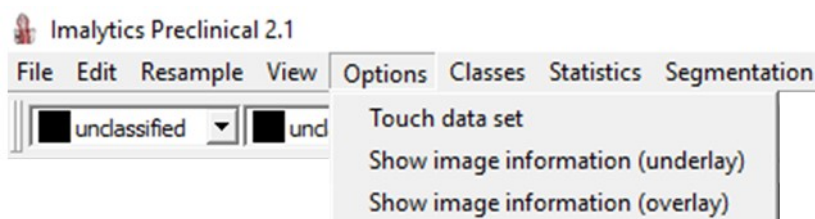
View > Set background color: Changes the background color in slice views.

View > Find brightest pixel (underlay): Sets the viewing center to the brightest pixel of the underlay.

View > Find brightest pixel (overlay): Sets the viewing center to the brightest pixel of the overlay.

Options

The options menu provides the following functionality:



Options > Touch data set: This loads the entire data set into memory. Many file formats are read on demand which may cause many interruptions during some operations. Then it is better to once load the data set into memory.

Options > Show image information (underlay): Show the meta-data of the underlay, i.e. information about the voxel sizes and more.

Options > Show image information (overlay): Show the meta-data of the overlay, i.e. information about the voxel sizes and more.

Options > Set menu font size: Adjust the font size of the menu.

Options > Set tool bar icon size: Adjust the icon size of the tool bar.

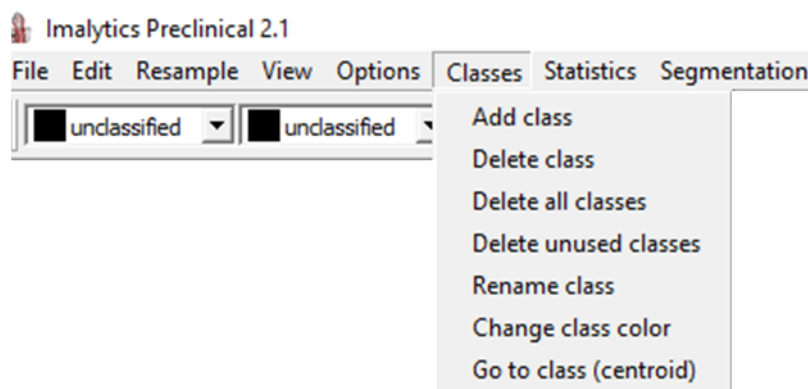
Options > Set language...: Adjust the language of the menu.

Options > Set language to English. Set language back to English.

Options > Toggle console. Activate a console window to see some text output.

Classes

The Classes menu allows the user to add, delete, or rename classes or to change their colors.



Classes > Add class: Adds a new class with a user-provided name. The class will get a certain color based on a lookup table, e.g. blood will be red and kidney will be yellow. Classes with names such as red, green, blue, or magenta will get the corresponding colors (based on the color list in https://en.wikipedia.org/wiki/Lists_of_colors).

Classes > Delete class: Deletes a single class. The user needs to select which class.

Classes > Delete all classes: Deletes all classes. The segmentation is cleared.

Classes > Delete unused classes: Deletes all classes which have no voxel assigned to them.

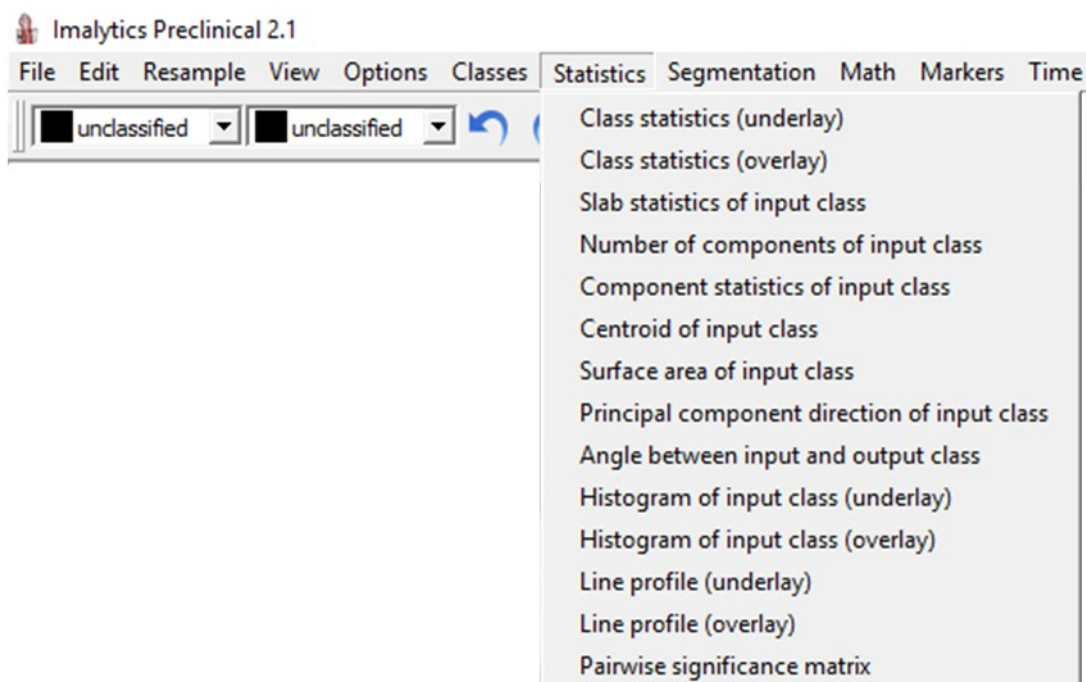
Classes > Rename class: Renames a class given a user-provided name. This will not change the color though.

Classes > Change class color: Opens a color dialog to select a new color for a class.

Classes > Go to class (centroid): Sets the viewing center (the red cross) to the centroid of the user-selected class. This is useful to find a small class. The voxel with the centroid may have another class, however, e.g. if the class has a ring shape.

Statistics

The Statistics menu offers functionality to quantify parameters of the images using the segmentation.



Statistics > Class statistics (underlay)

The following parameters are computed for each class:

- Volume (in mm³)
- Mean intensity (same unit as the image intensity, typically a concentration unit)
- Total (product of volume and mean intensity; if the mean is a concentration, the total value is an amount)
- Stddev (Standard deviation of the intensities of the class)
- Min (minimum value of all intensities in the class)
- Max (maximum value of all intensities in the class)

The data is stored in an excel sheet (.csv file). For proper open the file in Excel, the “Region and Language” settings of the operating systems need to be set to English (Figure 5). The excel sheet also contains the time index and time which is only relevant for 4D images, however.

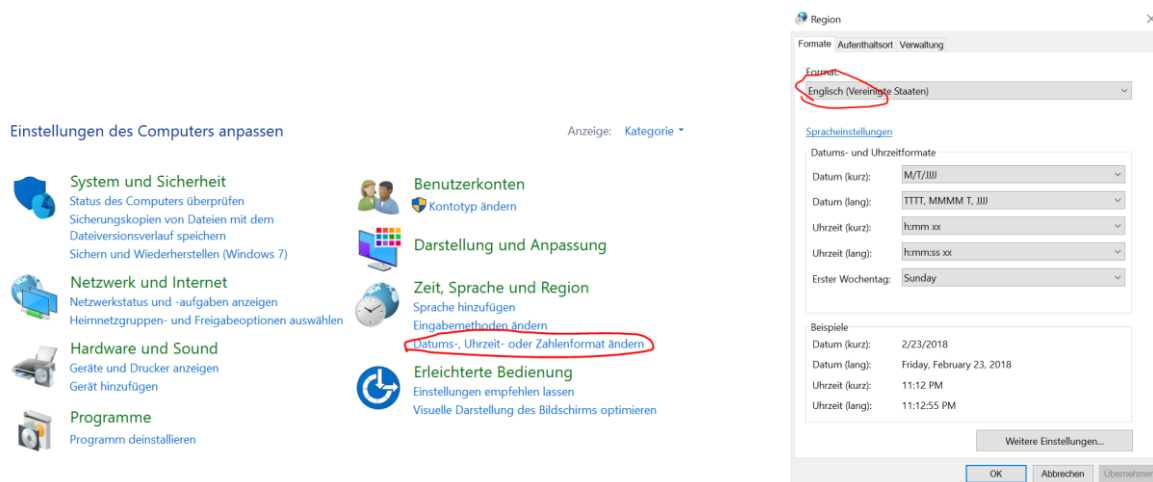


Figure 5: Statistics are saved as .csv files. The format in the „Region and Language“ settings of the operating system (e.g. Windows 10 64bit) needs to be set to English. Otherwise Excel will not read the numbers correctly.

Statistics > Class statistics (overlay): Computes an excel sheet similar to **Statistics > Class statistics (underlay)**, except that the overlay is used for quantification. This can be used to quantify the fluorescence in segmented organs or lesions using μ CT-FMT scans [9]–[11].

Statistics > Slab statistics of input class: Computes statistics of the input class for each slice (or thicker slabs) and stores these in an excel sheet. You can enter a slab thickness, e.g. 10 voxels. Slabs are considered in the Z direction. For each slab, the volume of the input class is computed as well as the number of components. Furthermore, you can enter a histogram (i.e. the borders such as 1.0;2.0;3.0;4.0). The components are sorted into the histogram buckets based on the provided histogram borders (e.g. 5 buckets for 4 borders) based on the component volume. The resulting .csv file contains one row for each slab and the row includes the histogram. The histograms can be used to analyze blood vessels, e.g. to find out if the number of small or large blood vessels are increased under some condition.

Statistics > Number of components of input class: Computes the number of disconnected components in the input class. This can be used to count the number of metal fragments for example [58].

Statistics > Component statistics of input class: Computes statistics of all disconnected components of the input class and saves these as .csv file. The list is sorted by volume (descending). The .csv file contains one row for each component with the number of voxels, volume, and the XYZ-coordinates of the centroid. This can be used to analyze number and volumes of lung metastases in μ CT scans.

Statistics > Centroid of input class: Computes the 3D position of the centroid of the input class.

Statistics > Voxel surface area of input class: Computes the surface area of the input class by summing all the areas of the boundary surfaces of boundary voxels.

Statistics > Principal component direction of input class: Computes the dominant 3D direction of the input class by computing the dominant eigenvector of the covariance matrix of all segmented voxel positions.

Statistics > Angle between input and output class: Computes the angle of input and output classes by computing their principal component directions. The can be used to quantify the angle between two bones or other classes [58].

Statistics > Histogram of input class (underlay): Computes the histogram of the underlay voxel values of the input class. The user provides the range of the histogram and the number of buckets. The histogram is saved as .csv file.

Statistics > Histogram of input class (overlay): Computes the histogram of the overlay voxel values of the input class.

Statistics > Line profile (underlay): Computes a line profile through the underlay. The line goes through the viewing center (red cross) in the x-direction (data space). The full width half max is computed for the local maximum next to the viewing center.

Statistics > Line profile (overlay): Computes a line profile through the overlay instead of the underlay (see above).

Statistics > Pair wise significance matrix: Computes a pairwise significance matrix which visualizes the pairwise significances of a multi group test, e.g. one-way-ANOVA with a Tukey post-test. This feature requires Graphpad Prism to generate a one-way-ANOVA with a Tukey post-test. Then copy the summary column of the post-test into the clipboard. Then Press *Menu > Statistics > Pairwise significance matrix* which generates an image an places it into the clipboard. Open MS Powerpoint for example and paste the matrix there (Figure 6).

Statistics > Dice scores against other segmentation: It requires a segmentation. Asks the user to select another segmentation file. Then the dice score is computed for each class that exists in both segmentation. The dice score represents the similarity, e.g. 1 means perfect match and 0 means no overlap at all. A .csv file is generated which contains all dice scores for these segmentations.

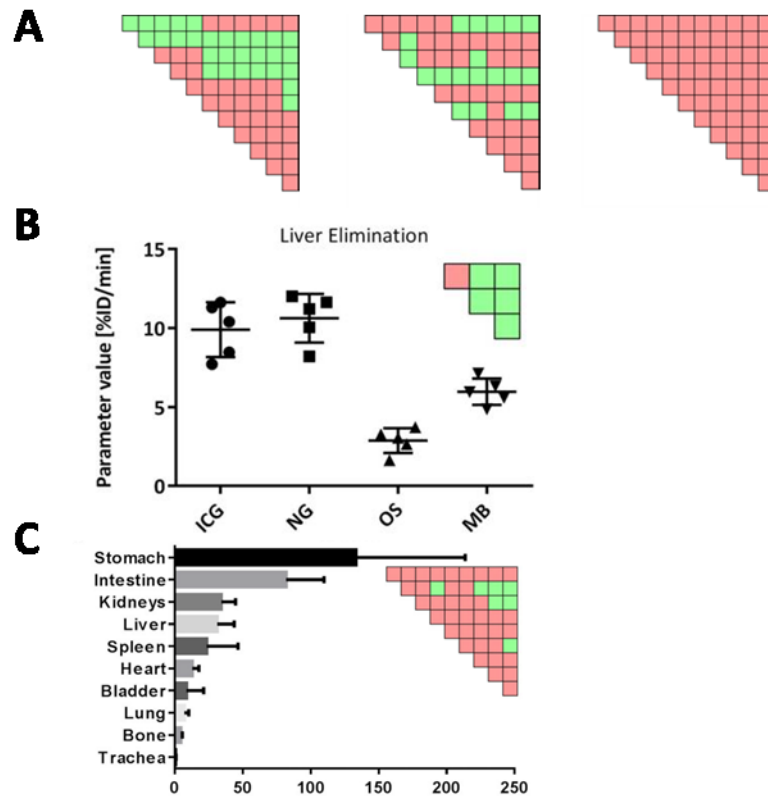
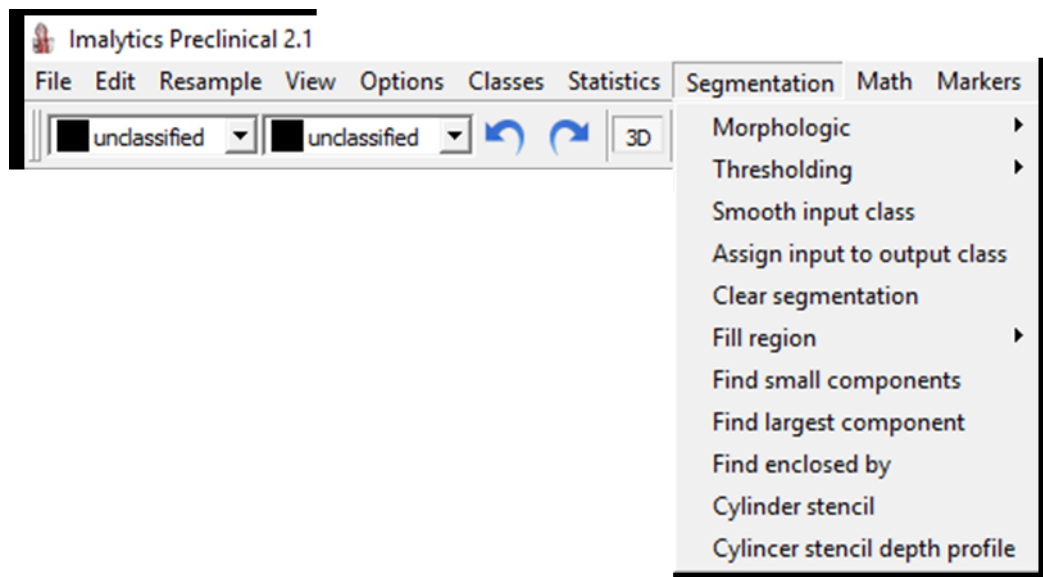


Figure 6: Examples of pairwise significance matrix (PSM). (A) Each box in the PSM represents the comparison of two groups. Significant differences ($p < 0.05$) between pairs of groups are shown in green and insignificant differences in red. (B) Example of PSM from the publication Al Rawashdeh et al. 2017 [19]. (C) Example of PSM from the publication Rosenhain et al. 2016 [12].

Segmentation

This menu provides functionality to alter the segmentation. The operations may use input and output classes.



Segmentation > Morphologic: Applies different morphologic operations on the input class. These are implemented using a distance map. The user has to enter a value to define the extent of the operation.

- > **Dilate** enhances makes a class “thicker” by extending the class to the outside. Enter a value for the dilating distance.
- > **Erode** shrinks a class from the borders. The removed voxels are put into the output class. This can be used to segment a region into “onion peelings”, e.g. the rim and core of a tumor.
- > **Open** performs Erode and Dilate successively, which removes small islets.
- > **Close** performs Dilate and Erode successively which fills small holes.
- > **Coat** is like **Dilate** except that the added voxels are assigned to the output class.

Segmentation > Thresholding: Provides different threshold operations. These may depend on one or two user-provided thresholds. The thresholding operations make use of the underlay voxel intensities.

- > **Above** assigns all voxels with intensity above (and equal) to a threshold to the output class.
- > **Below** assigns all voxels with intensity below (and equal) to a threshold to the output class.
- > **Range** asks the user for two thresholds (upper and lower bound).
- > **Above center** uses the current viewing center value as threshold. This is the same threshold that is used for the 3D iso-rendering of the underlay.

Segmentation > Smooth input class: Performs a smoothing operation on the input class. This is implemented using a Gaussian filter with standard deviation of 2 (voxels) and subsequent thresholding.

Segmentation > Assign input class to output class: Changes all voxels from the input class to the output class.

Segmentation > Clear segmentation: Changes all voxels to the class “unclassified”.

Segmentation > Fill region: Provides several region filling (or region growing) operations. These are seed point-based operations, i.e. the start at a seed point (the active point set by a right click). Then they fill the six-connected region with the output class.

- > **Fill region unlimited** performs an unbounded filling operation.

- > **Fill region volume-bounded** stops when a user-provided volume is reached.

- > **Fill region distance-bounded** stops when a user-provided shortest path distance to the seed point is reached.

- > **Fill region again** repeats the previous fill region operation.

- > **Grow into output class** grows the input class by one voxel into the output class, i.e. is not based on the seed point.

Segmentation > Find small components: Finds all six-connected components of the input class below a user-provided volume and assigns these to the output class.

Segmentation > Find largest component: Finds the largest six-connected component and assigns it to the output class.

Segmentation > Find enclosed by: Finds all 6-connected components of the input class enclosed by a user-provided class and assigns these to the output class. The user can provide a minimal enclosing ratio (between 0 and 1). This is useful for components that are connected to the outside, e.g. the trachea which never reach 1 because the border-surface is added to the denominator only.

Segmentation > Create region:

Creates simple segmentations of fixed shape. This is useful to estimate the brightness or noise in a well-defined region. The active point (green cross-hair) is the center of the generated region.

- > **Cylinder:** Generates a cylinder with user-defined length and diameter.

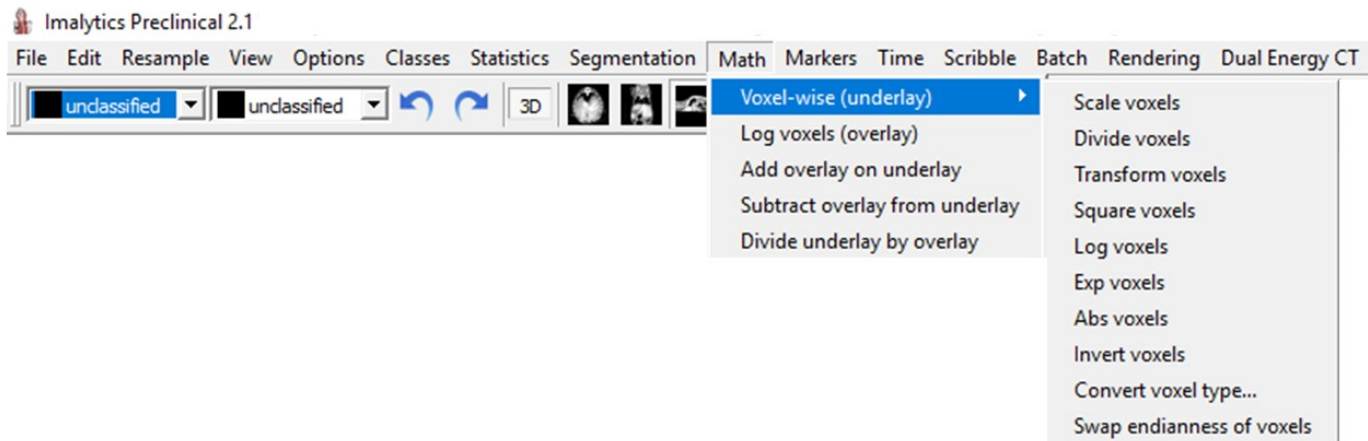
- > **Sphere:** Generates a sphere with user-defined diameter.

- > **Box:** Generates a box-shaped region with user-defined size.

Segmentation > Transfer segmentation to overlay: Transfers the underlay segmentation to the overlay space and saves it as a .seg file.

Math

This menu provides functionality to apply mathematical functions on the voxels. The operations are performed on the underlay unless otherwise noted. The operations are performed in memory and do not affect the file that has been opened to load the underlay or overlay.



Math > Voxel-wise (underlay): Performs the same operation to each voxel.

- > **Scale voxels:** Applies a user-provided scale factor to each voxel.
- > **Divide voxels:** Applies a user-provided denominator to each voxel, i.e. each voxel intensity is divided by the user-provided value.
- > **Transform voxels:** Applies a user-provided scale factor and additive offset to each voxel, i.e. the formula $y=ax+b$ is applied to each voxel whereby a is the scale factor and b is the offset.
- > **Log Voxels:** Applies the natural logarithm to each voxel. The voxel type is converted into float.
- > **Exp voxels:** Applies the exponential function to each voxel. The voxel type is converted into float.
- > **Abs voxels:** Applies the absolute operation, i.e. the sign is removed.
- > **Invert voxels:** Applies the inverse function to each voxel. The voxel type is converted into float. Zero values cannot be inverted and stay 0.
- > **Convert voxel type...:** Converts the voxel type into a user-provided type, i.e. char, uchar, short, ushort, int, uint, float or double. Values that are outside the range are clamped to the range, e.g. 1230 becomes 255 when converting to uchar which has range [0,255].
- > **Swap Endianness of voxels:** Converts little endian to big endian and vice versa, for each voxel. Some Linux systems provide big endian voxels.
- > **Zero lower bits:** Requires a short data set, e.g. from μ CT. Rounds the voxel values to a multiple of 2, 4, 8, depending on the number of bits entered by the user. This causes the lower bits to become zeros and the image can be compressed much stronger, e.g.

as .gff file. The lower bits are noisy anyways, e.g. for μ CT 5 bits can be zeroed with hardly any visible effect.

Math > Log voxels (overlay): Applies the natural logarithm to each voxel of the overlay. The voxel type is converted into float.

Math > Add overlay on underlay: Adds the overlay to the underlay. This requires that overlay and underlay have the same dimensions.

Math > Subtract overlay from underlay: Subtracts the overlay from the underlay. This requires that overlay and underlay have the same dimensions.

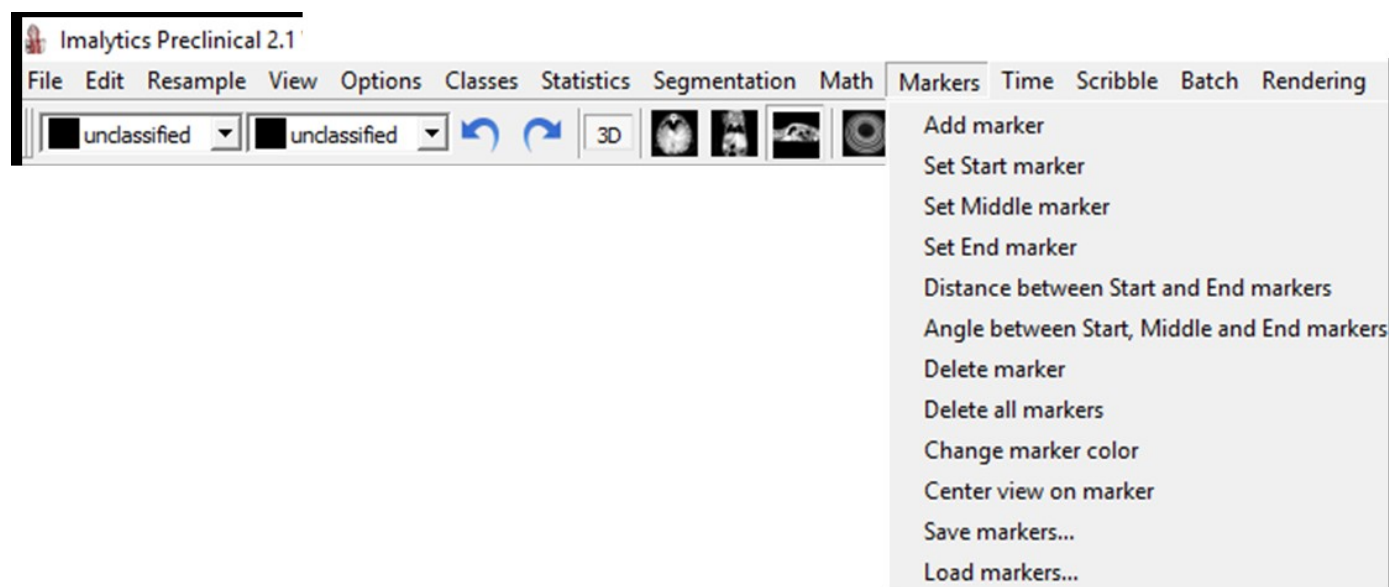
Math > Divide underlay by overlay: Divides the underlay voxels by the overlay voxels (only for non-zero overlay voxels). This requires that overlay and underlay have the same dimensions.

Math > HU calibration: Performs HU calibration on the underlay. Requires *Air* and *Water* classes with nonzero size. Computes mean intensity of *Air* and *Water* and applies scale and offset so that *Air* has intensity -1000 and *Water* has intensity 0.

Math > SUV calibration (overlay): Performs SUV calibration on the overlay. Overlay values are assumed to be given in [Bq/ml]. You will be asked to provide the injected dose in [Bq] and body weight in [g]. After calibration, the unit of the voxel values is [g/ml].

Markers

The Markers Menu provides functionality to manually set named markers at arbitrary positions. These can be used to compute distances (between Start and End markers), to initialize some path-based methods which require start and end points, or for marker-based fusion (see **Fusion** menu). Markers can be created in the underlay or overlay.



Markers > Add marker: Adds a new marker with a user-provided name at the active point (green cross).

Markers > Markers on overlay: If this is activated, the markers will be added to the overlay, otherwise to the underlay. In the 2D and 3D visualization either underlay or overlay markers are visualized, depending on this state.

Markers > Set Start marker: Adds a new marker named “Start” with magenta color at the active point (green cross).

Markers > Set Middle marker: Adds a new marker named “Middle” with cyan color at the active point (green cross).

Markers > Set End marker: Adds a new marker named “End” with cyan color at the active point (green cross).

Markers > Distance between Start and End markers: Computes the line distance (in mm) between Start and End markers.

Markers > Angle between Start, Middle and End markers: Computes the angle between Start, Middle and End markers.

Markers > Delete marker: Deletes one marker which has to be selected by the user.

Markers > Delete all marker: Deletes all markers.

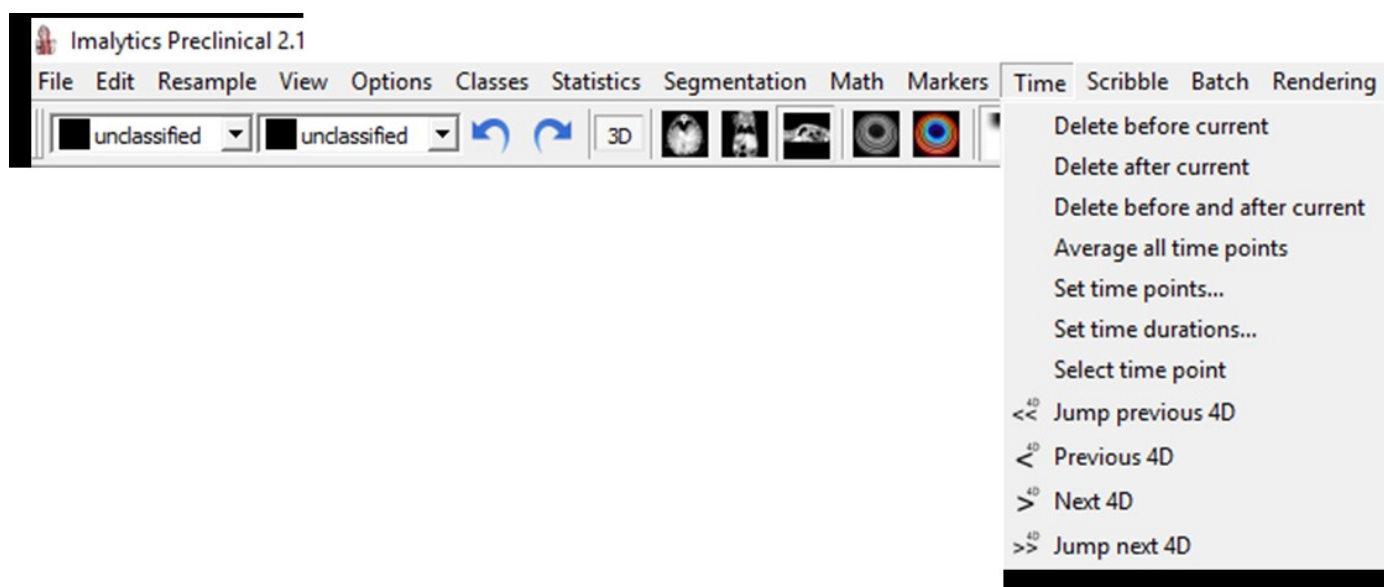
Markers > Center view on marker: Centers the view (red cross) on one marker selected by the user.

Markers > Save markers....: Saves the markers as comma-separated .csv file.

Markers > Load markers....: Loads a comma-separated .csv file to restore the markers.

Time

The time menu provides functionality for 4D data sets. The fourth dimension is the time.



Time > Delete before current: Deletes all time points before the current time point.

Time > Delete after current: Deletes all time points after the current time point.

Time > Delete before and after current: Deletes all time points before and after the current time point.

Time > Average all time points: Computes the average image of all time points and loads this image as underlay.

Time > Set time points...: Asks the user to enter the time points of the 4D data sets. These must be entered as one string where the numbers are separated by a semi-colon, e.g. "0.0;1.0;4.0".

Time > Set time durations...: Asks the user to enter the durations of the time points of the 4D data sets. These must be entered as one string where the numbers are separated by a semi-colon, e.g. "1.0;1.0;2.0".

Time > Find brightest time point: Goes to the time point with brightest pixel value. This method is based on the underlay if it has more than one time points, otherwise the overlay.

Time > Select time point: Goes to the time point provided by the user.

Time > Jump previous 4D: Jump 10 time points back.

Time > Previous 4D: Go one time point back.

Time > Next 4D: Go to the next time point.

Time > Jump next 4D: Jump 10 time points forward.

Time > Save time points to sub folders: Save each time point of a 4D image to a separate sub folder. The root folder and file name is provided by the user. This function is useful for cases where a segmentation is needed for each time point. The split data can be processed with the batch menu.

Channels

The channels menu provides functionality for 5D data sets. The fifth dimension represents the channels. This menu requires a special license.

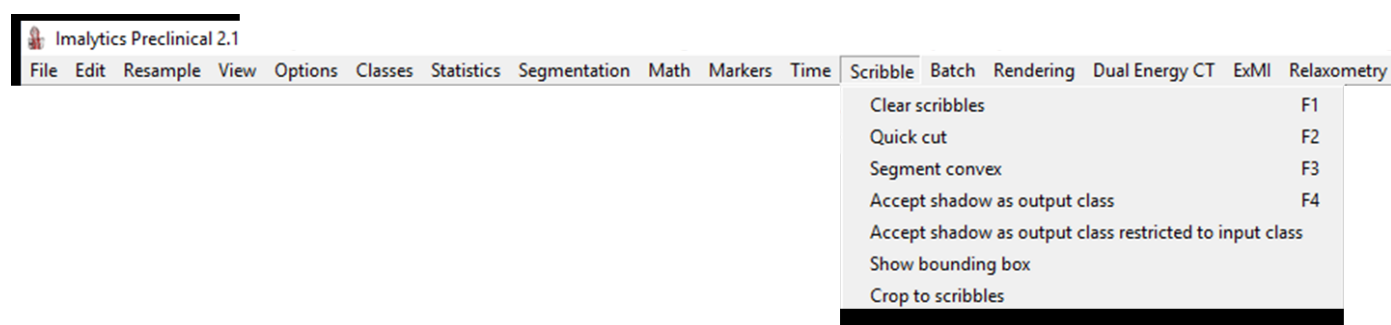
Channels->Average all channels: Computes the average image of all channels and loads this image as underlay.

Channels->Previous channel: Selects the previous channel as current channel.

Channels->Next channel: Selects the next channel as current channel.

Scribbles

The scribble menu provides functionality related to the "scribbles". Scribbles are drawn by the user into the slice views by the user to delineate regions. They appear as green voxels. Each voxel of the underlay can be marked, i.e. the scribbles are stored as a data set of the same X, Y and Z dimensions as the underlay. The scribbles are temporary and do not directly affect the segmentation upon being drawn.



Scribble > Clear scribbles (F1): Deletes all scribbles. This should be pressed before starting to delineate a region to delete some old scribbles that may exist in other slices.

Scribble > Quick cut (F2): Determines the convex hull defined by the scribbles and sets all voxels inside to unclassified.

Scribble > Segment convex (F3): Computes the convex hull defined by the scribbles. This is shown as semi-transparent red mask, called “shadow”.

Scribble > Accept shadow as output class (F4): Sets all shadow voxels to the output class.

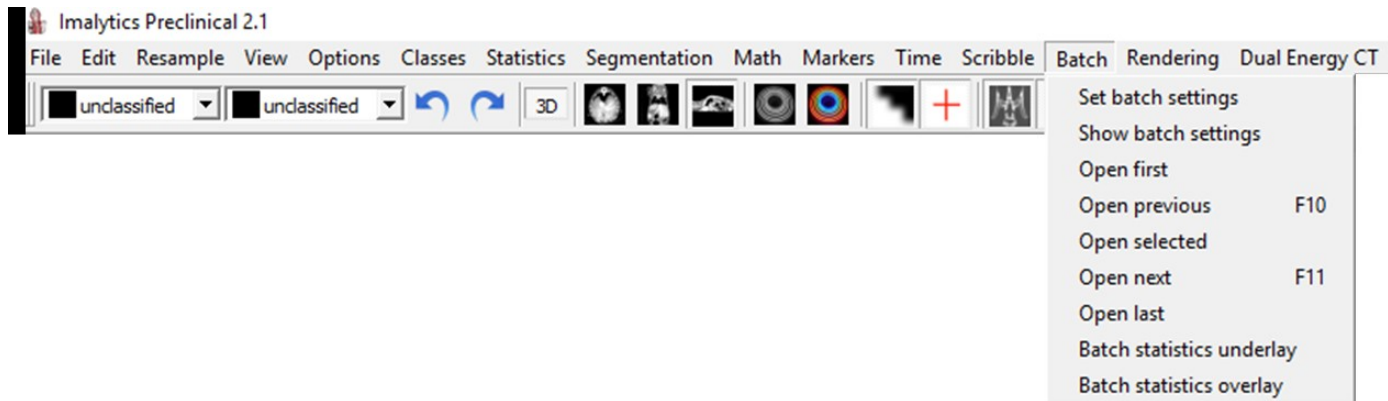
Scribble > Accept shadow as output class restricted to input class: Sets all shadow voxels to the output class but this is restricted to the input class.

Scribble > Show bounding box: Computes the bounding box around all scribbles. The bounding box is shown as semi-transparent red map called “shadow”.

Scribble > Crop to scribbles: Crops the data set to the bounding box around the scribbles.

Batch

The batch menu provides functionality to work on multiple data sets. These need to be stored in parallel folders. These data sets can represent multiple patients or mice for example, or multiple time points of the same mouse, as for longitudinal CT-FMT scans. The batch features are very useful to quickly get an overview of such files, and also to create consistent images of multiple subjects, i.e. under the same windowing settings.



Batch > Set batch settings: Captures the current settings for the batch operations and reports these to the user. These settings include the names of the underlay, overlay and segmentation files. Furthermore, the windowing settings, color tables and 3D camera position are captured. Also, a list of parallel subfolders is shown to the user.

Batch > Show batch setting: Shows the batch settings as described above.

Batch > Open first: Opens the data in the first subfolder.

Batch > Open previous (F10): Opens the data in the previous subfolder.

Batch > Open selected: Opens one data set of the batch that is selected by the user.

Batch > Open next (F11): Opens the data in the next subfolder.

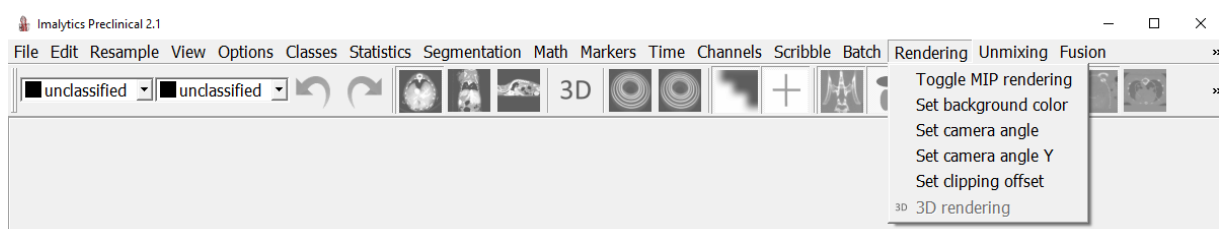
Batch > Open last: Opens the data in the last subfolder.

Batch > Batch statistics underlay: Computes the class statistics as described for the statistics menu. They are computed for all subfolders, however, and written into one .csv file. This can save a lot of effort, because often the same statistics need to be computed for many scans and get merged into one .csv file anyways. The statistics are computed on the underlay.

Batch > Batch statistics overlay: Computes the class statistics as described above but for the overlay.

Rendering

The rendering menu allows to configure the 3D rendering.



Rendering > Toggle MIP rendering: Switch to maximum intensity projection rendering for the underlay, instead of iso-surface rendering.

Rendering > Set background color: Allows the user to specify a background color (e.g. white) for 3D rendering. The default is black.

Rendering > Set camera angle: Allows the user to change the camera angle in the horizontal direction. Default is 10°. The total angle is twice this value. The camera angle in the vertical direction depends on the size of the panel.

Rendering > Set camera angle (Y): Allows the user to change the camera angle in the vertical direction.

Rendering > Set clipping offset: Allows the user to enter a clipping distance, which is the distance from the center of the dataset to the clipping plane.

Rendering > 3D rendering: Activates the 3D rendering panel.

Unmixing

The Unmixing menu provides functionality for spectral unmixing. It requires multichannel data sets, i.e. at least two channels. Unmixing involves two steps. First the pure spectra need to be determined by selecting regions with pure signal of the classes.

This menu requires a special license.

Unmixing > Save class spectra (underlay): Requires a segmentation with at least two classes. Then the mean underlay intensity of each class in each channel is computed. The result is saved as .spectra file. You can open it with a text editor.

Unmixing > Save class spectra (overlay): Same as above but the mean intensities are computed on the overlay.

Unmixing > Spectral unmixing (underlay): Requires a multichannel image. Asks the user to load a .spectra file (see above). Performs spectral unmixing, i.e. computes the intensities for the classes for each voxel. The computation searches for nonnegative values for each class such that the predicted mixed image (using the spectra) resembles the measurements. The resulting multi-channel image (one channel per class) is loaded as underlay and can be stored to disk.

Unmixing > Spectral unmixing (overlay): As above but operates on the overlay.

Fusion

Provides operations for fusing or aligning underlay and overlay. Both have a transformation (rotation and translation) to go between “data” and “world” coordinate systems. These transformations can be adjusted until underlay and overlay are aligned.

This menu requires a special license.

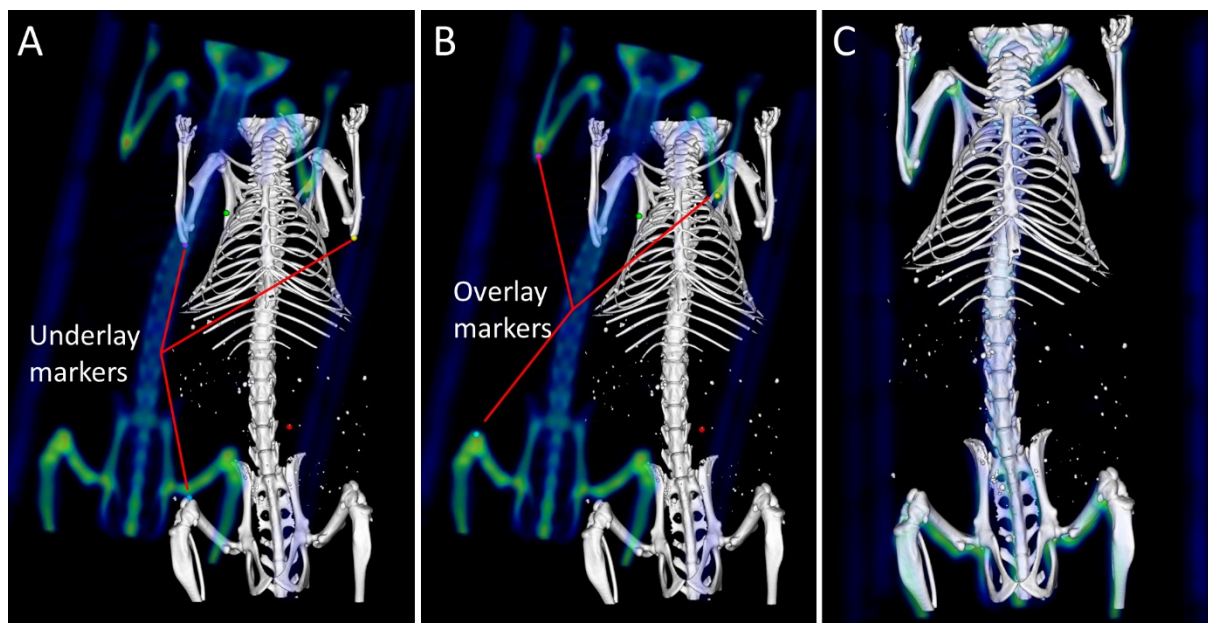


Figure 7. Marker-based fusion. Here, the underlay is a μ CT scan and the overlay is an artificial data set with blurry signal in the bones. A) Three markers are placed in the underlay at the two elbows and a knee. B) After toggling **Menu->Markers->Markers on overlay**, three markers are placed at corresponding positions and with identical names in the overlay. C) After clicking **Menu->Fusion->Register overlay to underlay (markers)**, the overlay transformation is adjusted and appears properly aligned to the underlay. At least three markers are required, and they should not be positioned on a line.

Fusion > Reset transform (underlay): Resets the underlay image transform to the identity transform.

Fusion > Reset transform (overlay): Resets the overlay image transform to the identity transform.

Fusion > Set translation (underlay): Asks for 3 values to update the underlay translation.

Fusion > Set translation (overlay): Asks for 3 values to update the overlay translation. This can be used to correct a shift in the fusion between underlay and overlay.

Fusion > Set rotation (underlay): Asks for 3 angles (X,Y,Z) to update the underlay rotation. The rotation is performed around the origin.

Fusion > Set rotation (overlay): Asks for 3 angles (X,Y,Z) to update the overlay rotation. The rotation is performed around the origin.

Fusion > Flip transform (underlay): Applies a flip in x, y, or z direction to the underlay transform. This can be used to generate a mirrored rendering of the underlay.

Fusion > Flip transform (overlay): Applies a flip in x, y or z direction to the overlay transform. This can be used to generate a mirrored rendering of the overlay.

Fusion > Center overlay on underlay: Sets the overlay transform so that the overlay is centered on top of the underlay.

Fusion > Register overlay to underlay (markers): Sets the overlay transform so that the residuum between corresponding underlay and overlay markers is minimized. Requires at least three markers in underlay and overlay with identical names. Markers are placed in the **Markers** menu.

Fusion > Register overlay to underlay (image-based, coarse): Computes the overlay transform so that the image difference (mean squared voxel differences) between underlay and transformed overlay, restricted to the input class, is minimized. The computation is performed on down-sampled

images and several initial transformations are used. If the result looks ok, the next menu item can be called to compute a more accurate transformation. If a good result is not found, you can try it again.

Fusion > Register overlay to underlay (image-based, fine): Similar to the menu item above (coarse fusion), however, the overlay transformation is computed at full resolution, if GPU memory allows. Large data sets (above 2 GB) are down-sampled anyways.

Bioluminescence

The Bioluminescence menu provides functionality for bioluminescence images.

This menu requires a special license.

Bioluminescence > Barrel correction: Applies barrel correction to the underlay, i.e. to each x-y-slice. This is also called rectification. The image is transformed by moving the pixels outward by the factor $1/(1+\alpha*r^2)$ where r is the distance of the pixel to the center and α is a camera-dependent coefficient (e.g. 0.00000025).

Bioluminescence > Vignetting correction: Applies vignetting correction to the underlay, i.e. to each x-y-slice. This is also called homogeneity correction. Each pixel is multiplied by a factor $\alpha * r^2 + 1$ where r is the distance of the pixel to the center. Alpha is a camera-dependent coefficient which also depends on the aperture.

Bioluminescence > Create example image: Generates an example image (1024x1024 pixels) showing a checker board with barrel distortion and vignetting. It can be corrected using **Bioluminescence > Vignetting correction** and **Bioluminescence > Barrel correction** with the default values.

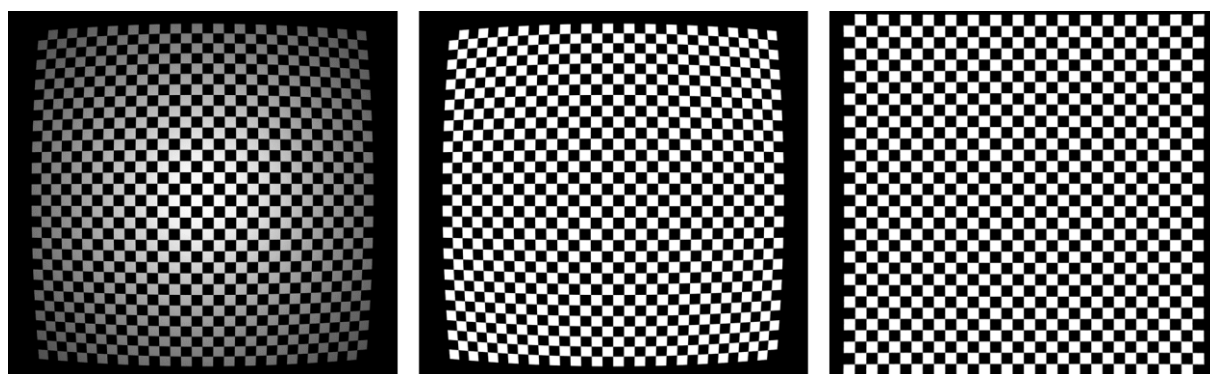


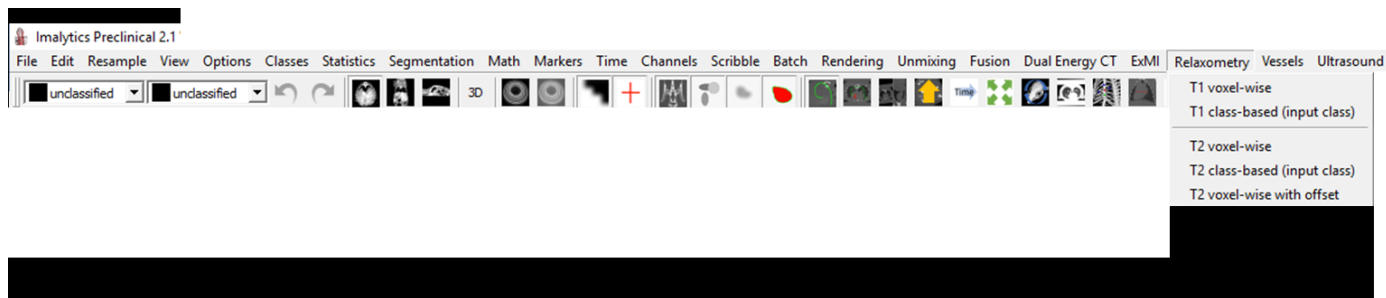
Figure 8: Vignetting and barrel correction. Left: barrel-distorted image with darker edges due to the vignetting effect. Middle: Vignetting-corrected image. Right: Barrel-corrected image.

Bioluminescence > Open Ivis data: Reads Ivis image data, i.e. a folder with excitation and emission images. The user needs to select a file named **AnalyzedClickInfo.txt**. The excitation image is loaded as underlay and the fluorescence image as overlay. Both are corrected for exposure time.

Relaxometry

This menu provides functionality for MRI relaxometry. It requires a 4D data set, i.e. 3D images at multiple time points (e.g. echo times). Relaxometry works by curve fitting, i.e. a parameterized curve is fitted to a measurement vector and the parameters are found by a cost function consisting of the squared residuum. During fitting, a small magnitude penalty ($0.001 * \text{measurements}^2 * R^2$) is added to the cost term to make the solution more stable.

This menu requires a special license.



Relaxometry > T1 voxel-wise: Fits an exponential curve $\beta \cdot \exp(-R1 \cdot t)$ to the voxel intensities over time. This requires that the time intervals of the 4D data set are properly set (see menu “Time”). The resulting R1 map, i.e. one value for each voxel, is loaded as overlay.

Relaxometry > T1 class-based (input-class): As above but the fitting is performed only for the mean values of the input class and not per voxel. Therefore, the result is not a voxel map; it is stored as .csv file instead which can be opened with Excel.

Relaxometry > T2 voxel-wise: Fits an exponential curve $\beta \cdot \exp(-R2 \cdot t)$ to the voxel intensities over time. The resulting R2 map, i.e. one value for each voxel, is loaded as overlay.

Relaxometry > T2 class-based (input-class): As above but the fitting is performed on the mean values of the input class. The results are saved as .csv file, which can be opened in Excel.

Relaxometry > T2 voxel-wise with offset: Fits an exponential curve $\beta \cdot \exp(-R2 \cdot t) + c$ to the voxel intensities over time. The resulting R2 map, i.e. one value for each voxel, is loaded as underlay. The offset is important for high R2 values (<http://www.ncbi.nlm.nih.gov/pubmed/23359410>). This feature is currently broken.

Vessels

This menu provides functionality to analyze blood vessels. This menu requires a special license.



Vessels > Set start point: Sets a marker (see marker menu) at the active point (right mouse click position) with the name “Start”.

Vessels > Set end point: Sets a marker (see marker menu) with the name “End”.

Vessels > Elastic sphere diameters: Fits an elastic sphere (Figure 9) through a segmented blood vessel [40]. Requires start and end points. Computes an excel sheet. Can be computationally expensive for large data sets and long paths.

Vessels > Path tortuosity: Computes the shortest path between start and end markers. Also computes the line distance. The ratio is the path tortuosity, i.e. tortuous paths will result in a larger score.

Vessels > Distance map inside input class: Computes a distance map [40] and loads it as overlay. This distance map contains the distances to the nearest voxel inside the input class.

Vessels > Distance map outside input class: Computes a distance map [40] and loads it as overlay. This distance map contains the distances to the nearest voxel of the input class.

Vessels > Create vascular example image: Creates a simple example image and loads it as underlay.

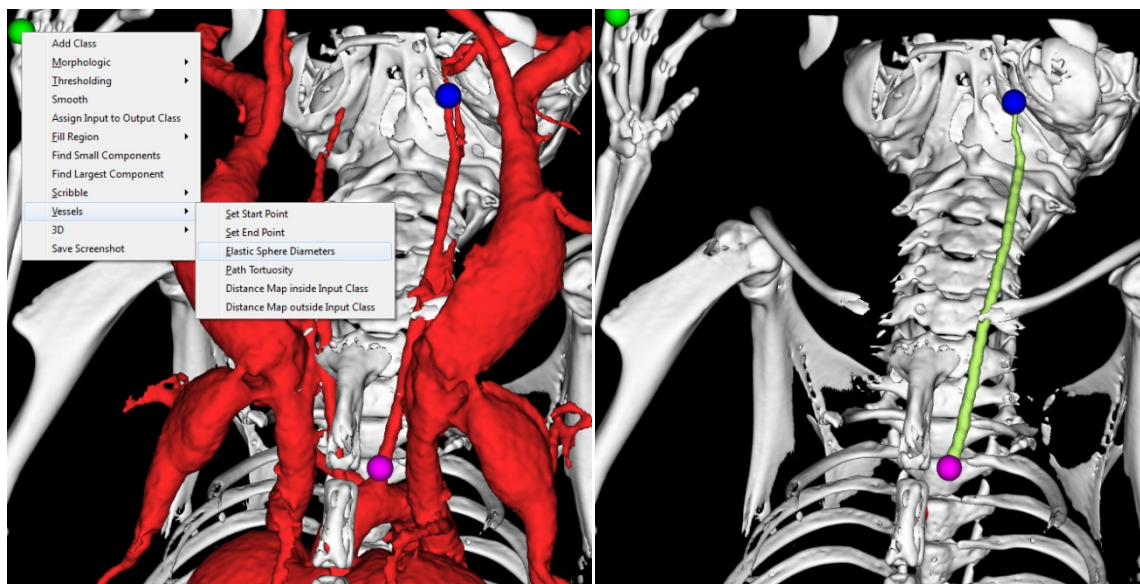


Figure 9: Blood vessel diameter measurements. A segmentation of blood vessels is required. Then the start (purple) and end (blue) points need to be set. They can be placed using the 3D view but it is better to position them in the center of the vessel using a slice-based view. The vessel diameters can be computed using Menu >Vessels > Elastic Sphere Diameters. The results are saved to a spread sheet file. The path is visualized (you can go back with Undo).

3D Printing

The menu item “3D Printing” allows conversion of the underlay or segmentation into .stl files which are required by many 3D Printers as input (Figure 10). These files contain many triangles describing a surface and can be inspected using the free software STLViewer (<http://sourceforge.net/projects/stlviewer/>). Currently, only one material (color) is supported. All segmented classes will be merged upon export. The 3D printing menu may not be available, depending on the license. **Currently, the printed result is mirrored as compared to the visualization. A work-around is to use Menu->Edit->Flip.**

This menu requires a special license.

3D Printing->Save underlay iso surface as .stl file: Saves the underlay iso contour, which is visualized in the 3D view, as .stl file.

3D Printing->Save entire segmentation as .stl file: Saves the complete segmentation (all classes), as one single-channel .stl file.

3D Printing->Save input class as .stl file: Saves the input class as .stl file.

3D Printing->Save .stl files as ASCII: If this is activated, .stl files will be saved as ASCII files. Otherwise as binary files with approximately a quarter of the size.

3D Printing->Create support below input class: Creates a new class “Support” and fills unclassified voxels below (in Z) the input class with support voxels. This filling stops at the height of the lowest voxel.

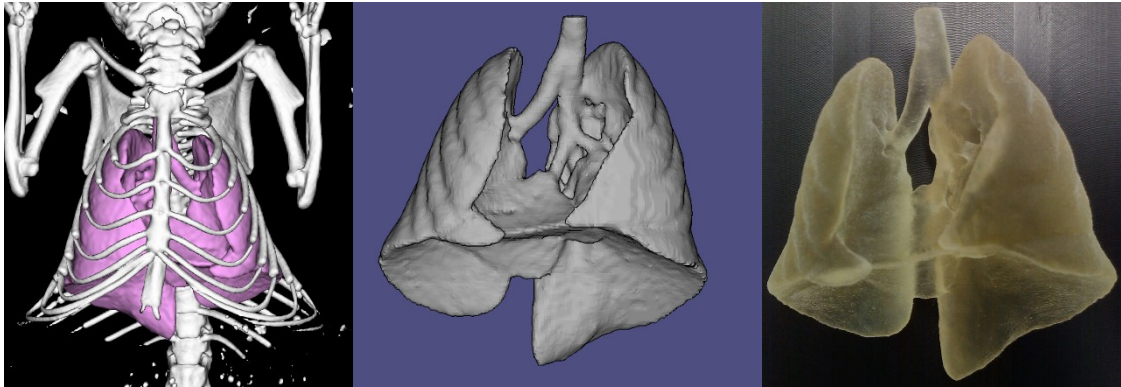


Figure 10: Left: CT volume and segmented lung. Middle: Lung segmentation exported as .stl file, visualized with STLViewer. Right: Printed 3D lung.

Macros

The menu item “Macros” provides special functionality, e.g. for segmentation of the mouse bed and body fat. This menu requires a special license.

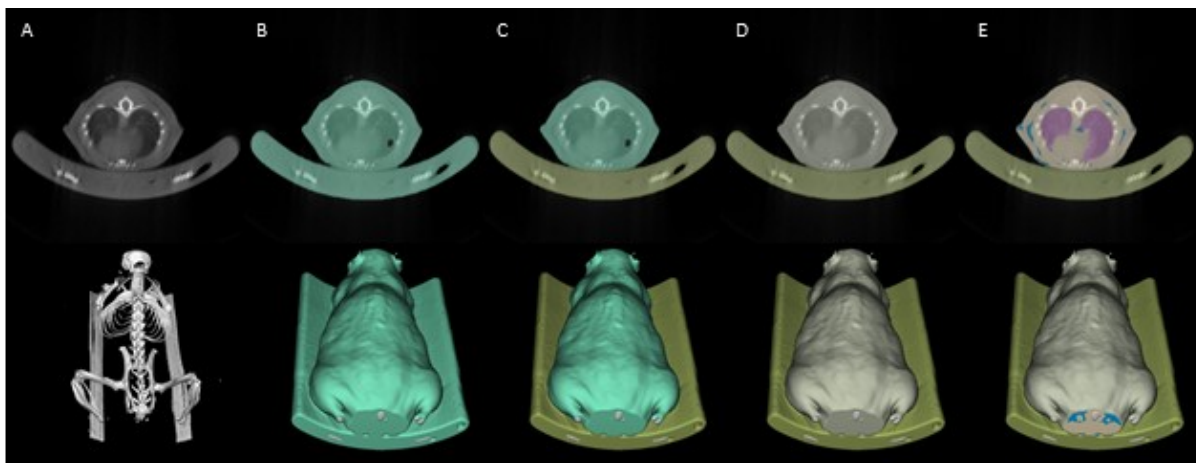


Figure 11. Steps used for mouse bed and body fat segmentation. A) First, the HU-calibrated μ CT scan is required (air at 0, water at 1000). B) Using thresholding (e.g. above 500) the class Matter is segmented. It might require cleaning, e.g. by smoothing. C) Using the macro described below, the round mouse bed on the bottom is segmented. D) Using another macro, the mouse body is segmented and small holes are filled. E) With another macro, the bones, lung and body fat are segmented.

Macros->Mouse bed (round, bottom, requires Matter class): Segments the bottom part of a round mouse bed. The user needs to enter the thickness of the round mouse bed. Requires class *Matter* containing matter (i.e. everything except air).

Macros->Mouse bed (round, top, requires Matter class): Segments the top part of a round mouse bed, similar as above.

Macros->Body (requires Bed and Matter classes): Segments the body of the mouse by searching the largest connected component of *Matter*. Then enclosing air regions are found and added to class *Body*.

Macros->Segment Bone, Lung and Fat (requires Body class): Segments *bone*, *lung* and *fat* classes based on the *Body* class. Requires HU-calibrated values (see Math>HU calibration). Asks the user for thresholds for bone, lung and fat. Below are some suggested thresholds for common scans.

Device	Scan mode	Bone threshold	Lung threshold	Fat threshold
Bruker 1278	Cont. 1mm filter	2000	600	850
CT-Imaging	HQD6565	2000	700	900
CT-Imaging	HQD4065	2000	600	900

Bone

The menu item “Bone” provides functionality to characterize bone structures. To meet widespread conventions, the bone statistics are provided in cm (or cm², etc.) and not in mm. Most of the methods below require an input class (can be unclassified) and a threshold. Furthermore, many parameters (e.g. the bone volume) are computed by oversampling the dataset (with linear interpolation) to improve the accuracy.

This menu requires a special license.

Bone > Create test data (sphere): Creates a 3D image containing a sphere. It reports the bone statistics of the sphere which can be checked using the functionality below.

Bone > Bone volume (BV) of input class: Computes the bone volume in [cm³], i.e. the volume of the region inside the input class which has intensity greater or equal to the threshold.

Bone > Total volume (TV) of input class: Computes the total volume in [cm³] of the input class.

Bone > Bone volume fraction (BV/TV) of input class: Computes the ratio of bone volume and total volume of the input class.

Bone > Bone surface (BS) of input class: Computes the surface area of the region with intensity higher than the threshold in the input class.

Bone > Bone-specific surface (BS/BV) of input class: Computes the ratio of BS and BV.

Bone > Bone mineral density (BMD) of input class: Computes the mean intensity of the input class. The unit is the same as the unit of the voxels. To achieve [g/cm³] the user needs to calibrate the intensities.

Bone > Trabecular thickness (Tb.Th) of input class: Computes the mean distance of the bone (i.e. the region higher than the threshold) to the outside of the bone. This is performed using a distance map. The operation is restricted to the input class.

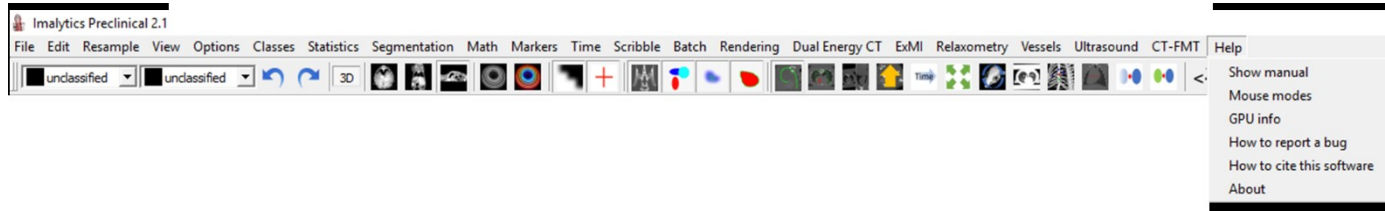
Bone > Trabecular separation (Tb.Sp) of input class: Computes the mean distance of the non-bone (i.e. the region lower than the threshold) to the inside of the bone. This is performed using a distance map. The operation is restricted to the input class.

Bone > Set crop extension: Allows the user to set a crop extension distance for the computations in voxels. This value extends the bounding box around the input class and has an effect on the resulting value. The value should be similar to the structural size, e.g. the trabecular separation.

Bone > Bone statistics (all) of input class: Computes all above-mentioned parameters for the input class and saves the result as a spread sheet file (.csv).

Help

The Help menu has the following entries:



Help > Show manual: Opens this manual in a pdf viewer.

Help > Mouse modes: Shows a list of the mouse modes.

Help > Show video tutorials (These videos have subtitles in English, German, Chinese and French which can be activated with the VLC player).

- **Scribbles:** Shows how to use the scribbles.
- **Toolbar:** Goes through the toolbar items and features.
- **Fill region:** Shows how to use fill region for segmentation.
- **Spectral unmixing:** Explains the spectral unmixing features.
- **Spectral unmixing MILabs:** Explains the spectral unmixing features of the MILabs device.

Help > GPU info: Shows information about the GPU.

Help > How to report a bug: Shows information how to best report a bug so that it can be found and fixed.

Help > How to cite: When using the software please mention the name of the software (Imalytics Preclinical) and cite the papers mentioned there.

Help > License info: Shows the license of the installed program, i.e. the remaining duration and which modules are activated.

Help > About: Shows information about the program.

5. GPU Processing

Many operations are run on the GPU since it has orders of magnitude more processing power. This program requires an NVIDIA GPU of compute capability 3.0 or higher (<https://developer.nvidia.com/cuda-gpus>).

GPUs have limited memory, which may cause problems for some operations (e.g. dilation) for large data sets which will be indicated by an error message with “cudaMalloc failed”. Solutions are to:

1. Order a GPU with more memory, e.g.
 - a. Any NVIDIA GPU with large memory and cuda compute capability 3.0 or higher.
 - b. Quadro K6000 12GB
 - c. Geforce Titan 6GB
 - d. Geforce 980 4GB
 - e. Geforce 970 4GB
 - f. Geforce 1030 2GB
 - g. Geforce 770 4GB
 - h. Geforce 680 4GB
 - i. Geforce 670 4GB
2. Close other applications that use GPU memory, e.g. other instances of this program.
3. Reduce the image size by cropping or down-sampling
4. Install the latest version where the situation might be improved.

The current GPU version and memory situation can be shown using Menu->Help->GPU Info.

6. Frequently asked questions

How can I save a fly-through movie of my image?

First, open the context menu by clicking the right mouse button in one of the slice viewing panels. Choose the point **Save movie (fly through)**. This will create a .wmv movie flying through the slices. Select the distance in the data set [mm]. (negative values will invert the direction) and the duration of the movie [s]. When the movie is ready, a window with “Movie is rendered.” appears. You need to press OK, otherwise the software will not continue running.

7. Troubleshooting

How to provide feedback?

As for all software programs, a screen shot image of the error message will be very helpful for troubleshooting. Thus, in case of problems, in addition to a short description of the problem, please also include, if possible, the error file you are having trouble with and the screen image. Please send it to fgremse@gremse-it.com and include your preferred email address and phone number in the message.

8. References

- [1] F. Gremse, M. Stärk, J. Ehling, J. R. Menzel, T. Lammers, and F. Kiessling, “Imalytics Preclinical: Interactive Analysis of Biomedical Volume Data,” *Theranostics*, vol. 6, no. 3, pp. 328–341, 2016.
- [2] F. Gremse *et al.*, “Hybrid μ CT-FMT imaging and image analysis,” *J Vis Exp*, no. 100, p. e52770, Jun. 2015.
- [3] A. Rix *et al.*, “Influence of repetitive contrast agent injections on functional and molecular ultrasound measurements,” *Ultrasound Med Biol*, vol. 40, no. 10, pp. 2468–2475, Oct. 2014.
- [4] B. Theek *et al.*, “Characterizing EPR-mediated passive drug targeting using contrast-enhanced functional ultrasound imaging,” *J Control Release*, vol. 182, pp. 83–89, May 2014.
- [5] S. C. Baetke *et al.*, “Squamous Cell Carcinoma Xenografts: Use of VEGFR2-targeted Microbubbles for Combined Functional and Molecular US to Monitor Antiangiogenic Therapy Effects,” *Radiology*, vol. 278, no. 2, pp. 430–440, Feb. 2016.
- [6] F. Fuge *et al.*, “In-vivo detection of the erythropoietin receptor in tumours using positron emission tomography,” *Eur Radiol*, Sep. 2014.
- [7] F. Düttenhoefer, M. E. Mertens, J. Vizkelety, F. Gremse, V. A. Stadelmann, and S. Sauerbier, “Magnetic resonance imaging in zirconia-based dental implantology,” *Clin Oral Implants Res*, Jun. 2014.
- [8] M. E. Mertens *et al.*, “Iron Oxide-labeled Collagen Scaffolds for Non-invasive MR Imaging in Tissue Engineering,” *Adv Funct Mater*, vol. 24, no. 6, pp. 754–762, Feb. 2014.
- [9] S. Kunjachan *et al.*, “Noninvasive optical imaging of nanomedicine biodistribution,” *ACS Nano*, vol. 7, no. 1, pp. 252–262, Jan. 2013.
- [10] S. Kunjachan *et al.*, “Passive versus active tumor targeting using RGD- and NGR-modified polymeric nanomedicines,” *Nano Lett.*, vol. 14, no. 2, pp. 972–981, Feb. 2014.
- [11] F. Gremse *et al.*, “Absorption reconstruction improves biodistribution assessment of fluorescent nanoprobe using hybrid fluorescence-mediated tomography,” *Theranostics*, vol. 4, no. 10, pp. 960–971, 2014.
- [12] S. Rosenhain, W. Al Rawashdeh, F. Kiessling, and F. Gremse, “Sensitivity and accuracy of hybrid fluorescence-mediated tomography in deep tissue regions,” *J Biophotonics*, Nov. 2016.
- [13] D. Doleschel *et al.*, “Erythropoietin improves the accumulation and therapeutic effects of carboplatin by enhancing tumor vascularization and perfusion,” *Theranostics*, vol. 5, no. 8, pp. 905–918, 2015.
- [14] W. Al Rawashdeh *et al.*, “Optical tomography of MMP activity allows a sensitive noninvasive characterization of the invasiveness and angiogenesis of SCC xenografts,” *Neoplasia*, vol. 16, no. 3, pp. 235–246, 246.e1, Mar. 2014.
- [15] Y. Shi *et al.*, “Fluorophore labeling of core-crosslinked polymeric micelles for multimodal in vivo and ex vivo optical imaging,” *Nanomedicine (Lond)*, vol. 10, no. 7, pp. 1111–1125, 2015.
- [16] C. Ergen *et al.*, “Targeting distinct myeloid cell populations in vivo using polymers, liposomes and microbubbles,” *Biomaterials*, vol. 114, pp. 106–120, Jan. 2017.
- [17] C. Hage *et al.*, “Comparison of the Accuracy of FMT/CT and PET/MRI for the Assessment of Antibody Biodistribution in Squamous Cell Carcinoma Xenografts,” *J. Nucl. Med.*, Aug. 2017.
- [18] Y. Tsvetkova *et al.*, “Balancing Passive and Active Targeting to Different Tumor Compartments Using Riboflavin-Functionalized Polymeric Nanocarriers,” *Nano Lett.*, vol. 17, no. 8, pp. 4665–4674, Aug. 2017.

- [19] W. Al Rawashdeh *et al.*, “Noninvasive Assessment of Elimination and Retention using CT-FMT and Kinetic Whole-body Modeling,” *Theranostics*, vol. 7, no. 6, pp. 1499–1510, 2017.
- [20] J.-M. Bangen *et al.*, “Targeting CCl₄ -induced liver fibrosis by RNA interference-mediated inhibition of cyclin E1 in mice,” *Hepatology*, vol. 66, no. 4, pp. 1242–1257, Oct. 2017.
- [21] N. Beztsinna, Y. Tsvetkova, M. Bartneck, T. Lammers, F. Kiessling, and I. Bestel, “Amphiphilic Phospholipid-Based Riboflavin Derivatives for Tumor Targeting Nanomedicines,” *Bioconjug. Chem.*, vol. 27, no. 9, pp. 2048–2061, Sep. 2016.
- [22] L. Novo *et al.*, “Decationized polyplexes as stable and safe carrier systems for improved biodistribution in systemic gene therapy,” *J Control Release*, vol. 195, pp. 162–175, Dec. 2014.
- [23] Y. Shi *et al.*, “Complete Regression of Xenograft Tumors upon Targeted Delivery of Paclitaxel via Π - Π Stacking Stabilized Polymeric Micelles,” *ACS Nano*, vol. 9, no. 4, pp. 3740–3752, 2015.
- [24] B. Theek *et al.*, “Sonoporation enhances liposome accumulation and penetration in tumors with low EPR,” *J Control Release*, vol. 231, pp. 77–85, Jun. 2016.
- [25] J. G. Mannheim *et al.*, “Standardization of Small Animal Imaging-Current Status and Future Prospects,” *Mol Imaging Biol*, Oct. 2017.
- [26] H. Dähring, J. Grandke, U. Teichgräber, and I. Hilger, “Improved Hyperthermia Treatment of Tumors Under Consideration of Magnetic Nanoparticle Distribution Using Micro-CT Imaging,” *Mol Imaging Biol*, vol. 17, no. 6, pp. 763–769, Dec. 2015.
- [27] J. Domey, U. Teichgräber, and I. Hilger, “Gold nanoparticles allow detection of early-stage edema in mice via computed tomography imaging,” *Int J Nanomedicine*, vol. 10, pp. 3803–3814, 2015.
- [28] S. Kossatz *et al.*, “Efficient treatment of breast cancer xenografts with multifunctionalized iron oxide nanoparticles combining magnetic hyperthermia and anti-cancer drug delivery,” *Breast Cancer Res.*, vol. 17, p. 66, May 2015.
- [29] M. Rose *et al.*, “ITIH5 mediates epigenetic reprogramming of breast cancer cells,” *Mol. Cancer*, vol. 16, no. 1, p. 44, Feb. 2017.
- [30] M. Brinkmann *et al.*, “Micro-computed tomography (μ CT) as a novel method in ecotoxicology--determination of morphometric and somatic data in rainbow trout (*Oncorhynchus mykiss*),” *Sci. Total Environ.*, vol. 543, no. Pt A, pp. 135–139, Feb. 2016.
- [31] C. Xie *et al.*, “Visualization of Vascular and Parenchymal Regeneration after 70% Partial Hepatectomy in Normal Mice,” *J Vis Exp*, no. 115, Sep. 2016.
- [32] C. Xie *et al.*, “Quantification of Hepatic Vascular and Parenchymal Regeneration in Mice,” *PLoS ONE*, vol. 11, no. 8, p. e0160581, 2016.
- [33] P. K. Srinivasan *et al.*, “Novel synthetic adhesive as an effective alternative to Fibrin based adhesives,” *World J Hepatol*, vol. 9, no. 24, pp. 1030–1039, Aug. 2017.
- [34] J. Kalder *et al.*, “New Large Animal Model for Aortic Aneurysms in the Viscerorenal Segment,” *J. Surg. Res.*, vol. 240, pp. 156–164, Mar. 2019.
- [35] J. Ehling *et al.*, “Micro-CT imaging of tumor angiogenesis: quantitative measures describing micromorphology and vascularization,” *Am. J. Pathol.*, vol. 184, no. 2, pp. 431–441, Feb. 2014.
- [36] J. Ehling *et al.*, “CCL2-dependent infiltrating macrophages promote angiogenesis in progressive liver fibrosis,” *Gut*, Feb. 2014.
- [37] J. Ehling *et al.*, “Quantitative Micro-Computed Tomography Imaging of Vascular Dysfunction in Progressive Kidney Diseases,” *J. Am. Soc. Nephrol.*, vol. 27, no. 2, pp. 520–532, Feb. 2016.

- [38] M. Bartneck *et al.*, “Histidine-rich glycoprotein promotes macrophage activation and inflammation in chronic liver disease,” *Hepatology*, vol. 63, no. 4, pp. 1310–1324, Apr. 2016.
- [39] W. Wei *et al.*, “Establishment of a rat model: Associating liver partition with portal vein ligation for staged hepatectomy,” *Surgery*, vol. 159, no. 5, pp. 1299–1307, May 2016.
- [40] F. Gremse *et al.*, “Virtual elastic sphere processing enables reproducible quantification of vessel stenosis at CT and MR angiography,” *Radiology*, vol. 260, no. 3, pp. 709–717, Sep. 2011.
- [41] M. Nazari-Jahantigh *et al.*, “MicroRNA-155 promotes atherosclerosis by repressing Bcl6 in macrophages,” *J. Clin. Invest.*, vol. 122, no. 11, pp. 4190–4202, Nov. 2012.
- [42] A. Schober *et al.*, “MicroRNA-126-5p promotes endothelial proliferation and limits atherosclerosis by suppressing Dlk1,” *Nat. Med.*, vol. 20, no. 4, pp. 368–376, Apr. 2014.
- [43] C. Schürmann, F. Gremse, H. Jo, F. Kiessling, and R. P. Brandes, “Micro-CT Technique Is Well Suited for Documentation of Remodeling Processes in Murine Carotid Arteries,” *PLoS ONE*, vol. 10, no. 6, p. e0130374, 2015.
- [44] S. Akhtar *et al.*, “Endothelial Hypoxia-Inducible Factor-1 α Promotes Atherosclerosis and Monocyte Recruitment by Upregulating MicroRNA-19a,” *Hypertension*, vol. 66, no. 6, pp. 1220–1226, Dec. 2015.
- [45] C. Schürmann *et al.*, “The NADPH oxidase Nox4 has anti-atherosclerotic functions,” *Eur. Heart J.*, vol. 36, no. 48, pp. 3447–3456, Dec. 2015.
- [46] K. T. Warzecha *et al.*, “Targeting and Modulation of Liver Myeloid Immune Cells by Hard-Shell Microbubbles,” *Adv Biosyst*, vol. 2, no. 5, May 2018.
- [47] B. Theek *et al.*, “Histidine-rich glycoprotein-induced vascular normalization improves EPR-mediated drug targeting to and into tumors,” *J Control Release*, vol. 282, pp. 25–34, 28 2018.
- [48] C. Ergen *et al.*, “Liver fibrosis affects the targeting properties of drug delivery systems to macrophage subsets in vivo,” *Biomaterials*, vol. 206, pp. 49–60, Jun. 2019.
- [49] X. Li *et al.*, “Activation of CXCR7 limits atherosclerosis and improves hyperlipidemia by increasing cholesterol uptake in adipose tissue,” *Circulation*, vol. 129, no. 11, pp. 1244–1253, Mar. 2014.
- [50] J. Gautheron *et al.*, “The necroptosis-inducing kinase RIPK3 dampens adipose tissue inflammation and glucose intolerance,” *Nat Commun*, vol. 7, p. 11869, Jun. 2016.
- [51] A. Assmann *et al.*, “The degeneration of biological cardiovascular prostheses under pro-calcific metabolic conditions in a small animal model,” *Biomaterials*, vol. 35, no. 26, pp. 7416–7428, Aug. 2014.
- [52] D. Dziewiecki *et al.*, “Osteoneogenesis due to periosteal elevation with degradable and nondegradable devices in Göttingen Minipigs,” *J Craniomaxillofac Surg*, vol. 44, no. 3, pp. 318–324, Mar. 2016.
- [53] C. Adamzyk *et al.*, “Bone tissue engineering using polyetherketoneketone scaffolds combined with autologous mesenchymal stem cells in a sheep calvarial defect model,” *J Craniomaxillofac Surg*, vol. 44, no. 8, pp. 985–994, Aug. 2016.
- [54] M. Kamal *et al.*, “Bone regeneration using composite non-demineralized xenogenic dentin with beta-tricalcium phosphate in experimental alveolar cleft repair in a rabbit model,” *J Transl Med*, vol. 15, no. 1, p. 263, Dec. 2017.
- [55] M. Kamal *et al.*, “A rabbit model for experimental alveolar cleft grafting,” *J Transl Med*, vol. 15, no. 1, p. 50, Feb. 2017.
- [56] L. Bohner, D. Habor, F. Gremse, P. Tortamano, S. Wolfart, and J. Marotti, “Accuracy of High-Frequency Ultrasound Scanner in Detecting Peri-implant Bone Defects,” *Ultrasound Med Biol*, vol. 45, no. 3, pp. 650–659, 2019.
- [57] M. Kamal *et al.*, “Bone regeneration in rabbit calvarial critical-sized defects filled with composite in situ formed xenogenic dentin and biphasic tricalcium

- phosphate/hydroxyapatite mixture,” *J. Biomed. Mater. Res. Part B Appl. Biomater.*, vol. 107, no. 3, pp. 773–782, Apr. 2019.
- [58] F. Gremse *et al.*, “Performance of lead-free versus lead-based hunting ammunition in ballistic soap,” *PLoS ONE*, vol. 9, no. 7, p. e102015, 2014.
 - [59] G. G. Yamoah *et al.*, “Data Curation for Preclinical and Clinical Multimodal Imaging Studies,” *Mol Imaging Biol*, Mar. 2019.
 - [60] S. Rosenhain, Z. A. Magnuska, G. G. Yamoah, W. A. Rawashdeh, F. Kiessling, and F. Gremse, “A preclinical micro-computed tomography database including 3D whole body organ segmentations,” *Sci Data*, vol. 5, p. 180294, 18 2018.
 - [61] M. Moreau *et al.*, “Priming the Abscopal Effect Using Multifunctional Smart Radiotherapy Biomaterials Loaded with Immunoadjuvants,” *Front Oncol*, vol. 8, p. 56, 2018.
 - [62] T. Opacic *et al.*, “Motion model ultrasound localization microscopy for preclinical and clinical multiparametric tumor characterization,” *Nat Commun*, vol. 9, no. 1, p. 1527, 18 2018.
 - [63] M. E. Mertens *et al.*, “USPIO-labeled textile materials for non-invasive MR imaging of tissue-engineered vascular grafts,” *Biomaterials*, vol. 39, pp. 155–163, Jan. 2015.
 - [64] D. Bauerschlag *et al.*, “Detection and Specific Elimination of EGFR+ Ovarian Cancer Cells Using a Near Infrared Photoimmunotheranostic Approach,” *Pharm. Res.*, vol. 34, no. 4, pp. 696–703, Apr. 2017.
 - [65] W. Wei *et al.*, “Intrahepatic Size Regulation in a Surgical Model: Liver Resection-Induced Liver Regeneration Counteracts the Local Atrophy following Simultaneous Portal Vein Ligation,” *Eur Surg Res*, vol. 57, no. 1–2, pp. 125–137, 2016.
 - [66] F. Gremse, A. Höfter, L. Razik, F. Kiessling, and U. Naumann, “GPU-Accelerated Adjoint Algorithmic Differentiation,” *Comput Phys Commun*, vol. 200, pp. 300–311, Mar. 2016.
 - [67] C. Sängler *et al.*, “Intrahepatic Vascular Anatomy in Rats and Mice--Variations and Surgical Implications,” *PLoS ONE*, vol. 10, no. 11, p. e0141798, 2015.