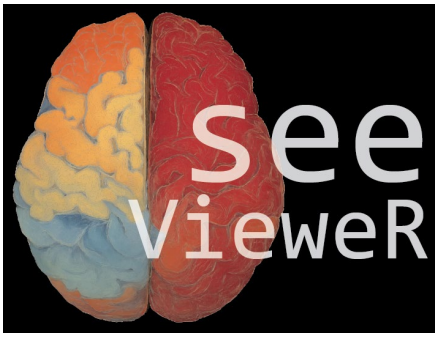
seeVieweR User Documentation

This document serves as a comprehensive guide for using the seeVieweR application — an advanced 3D/4D NIfTI viewer developed in MATLAB App Designer. It enables flexible visualization of NIfTI datasets, supports multiple overlays, and provides functionality, thresholding, transparency control, and high-resolution export. This guide includes detailed instructions and feature descriptions.



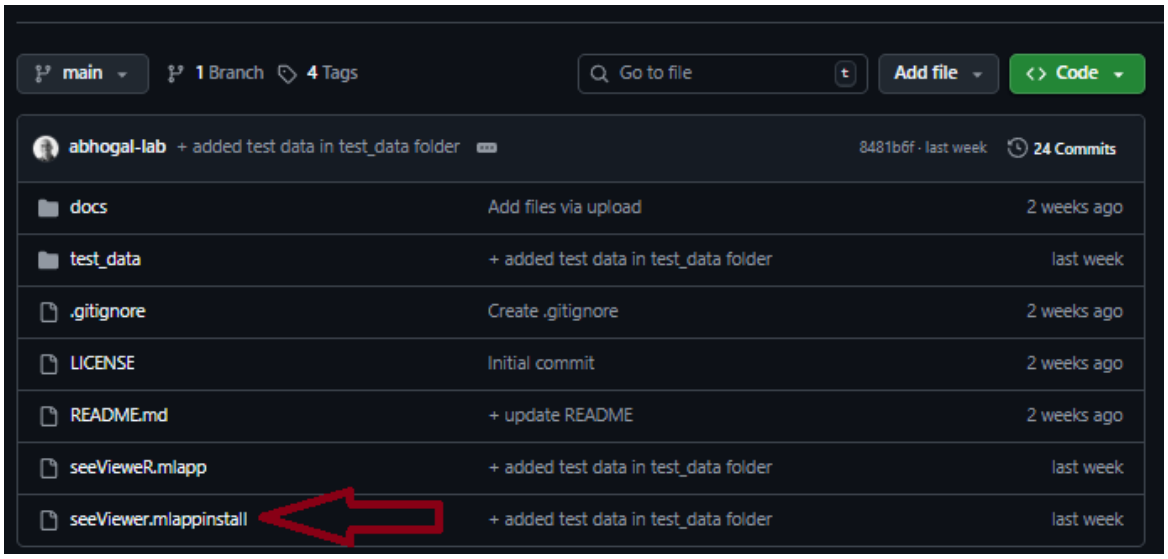
1. Introduction

The seeVieweR application provides an intuitive environment for exploring 3D and 4D data (NIfTI format). It combines volumetric rendering with interactive slice-based visualization and supports NIfTI file formats. For those using DICOM, conversion can be made using the DICOM-to- NIfTI – see dcm2nii

2. Installation and Setup

1. Download the latest release of seeVieweR from GitHub or MATLAB File Exchange.
2. Open MATLAB - R2023b is recommended since this was the development environment.
3. Install the app by double-clicking the `.mlappinstall` file or using the MATLAB Add-On Manager.
4. Launch seeVieweR from the Apps tab.
5. Ensure the following toolboxes are installed: Image Processing Toolbox, Signal Processing Toolbox.

Optional: Install FSL or ITK-SNAP for external verification of overlays. FSL is also handy for generating anatomical brain masks or segmentations.



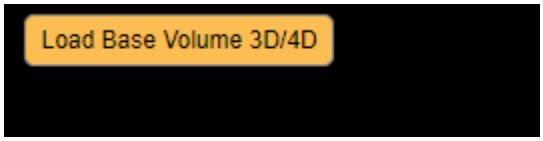
3. Loading Data

Click the 'Load Base Image' button and select a NIfTI file (.nii or .nii.gz). If a corresponding brain mask (e.g., *_brain_mask.nii*) is detected, it will be loaded automatically. Otherwise the user will also be allowed to load their own mask.

To add an overlay, click 'Load Overlay Image'. Overlays are interpolated to match the base image geometry, and you can choose between linear and nearest interpolation.

Notes: Assuming your image headers are intact, seeVieweR will automatically perform a registration of the overlay images and masks to the base image matrix. This is also important to consider when pressing the 'Export NIfTI' button since the exported images will have been resampled to the base image matrix.

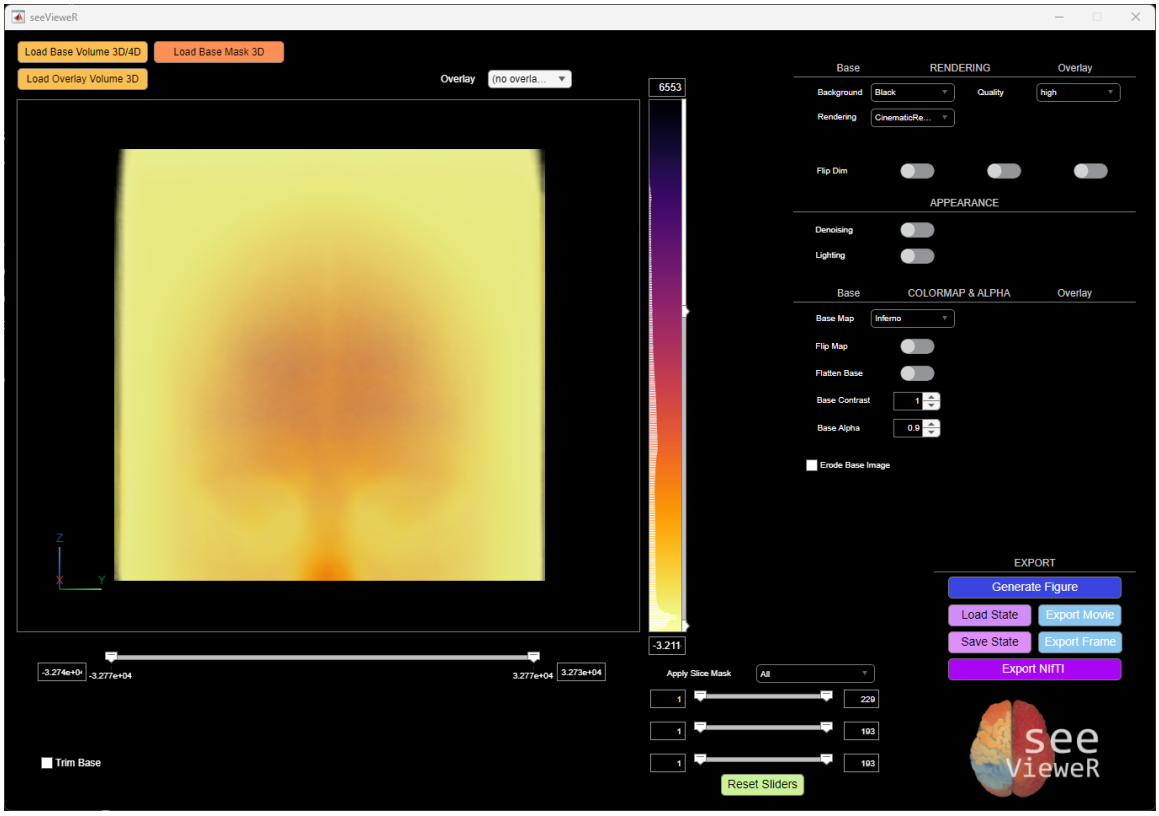
Try loading an example from the included test_data folder 'Anatomy.nii.gz'



The image will load in the viewer, however it will display as an opaque cube. This is because this image is non-masked and contains small non-zero values outside of the brain region.

4. Base Image Visualization

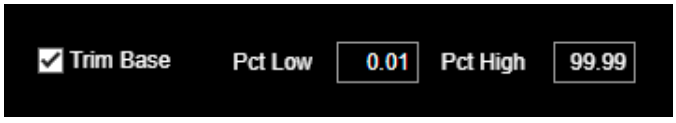
The base image forms the foundation of your visualization. You can adjust the intensity range using the Base Intensity Slider, which clips the displayed range without altering the underlying data. The colormap, opacity, and contrast can be modified using dedicated controls in the right-hand panel.



We can solve this in two ways (applies also to overlay images):

The first is by using the threshold slider under the image. This slider dictates the range of visible voxels. Moving the left slider will remove low value voxels and moving the right slider, high value voxels.

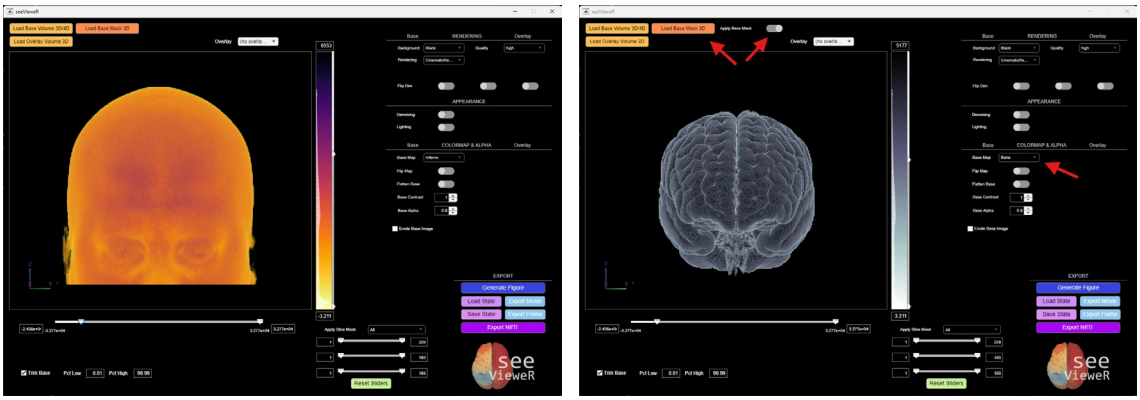
The second method is by using the 'Trim Base' option. When checking this box, an option to remove lower percentile and higher percentile voxels will appear. Thus only voxels between the defined lower and higher percentile values will be shown. When exporting, only the threshold slider will be applied to outputs.



If an image is masked, this step will not be needed. Alternatively you could load a mask and apply it.

Next, press the ‘Load Base Mask 3D’ button and then flip the ‘Apply Base Mask’ switch. You can also play with colormaps – apply the grayC or bone colormap in the ‘Base Map’ drop down in the right panel.

From here you can change rendering options, flip the image to match neurological versus radiological views and also apply the slice mask. These are the sliders located at the bottom of the viewer. As you will see later, these can be independently applied to the base image or the overlay image.



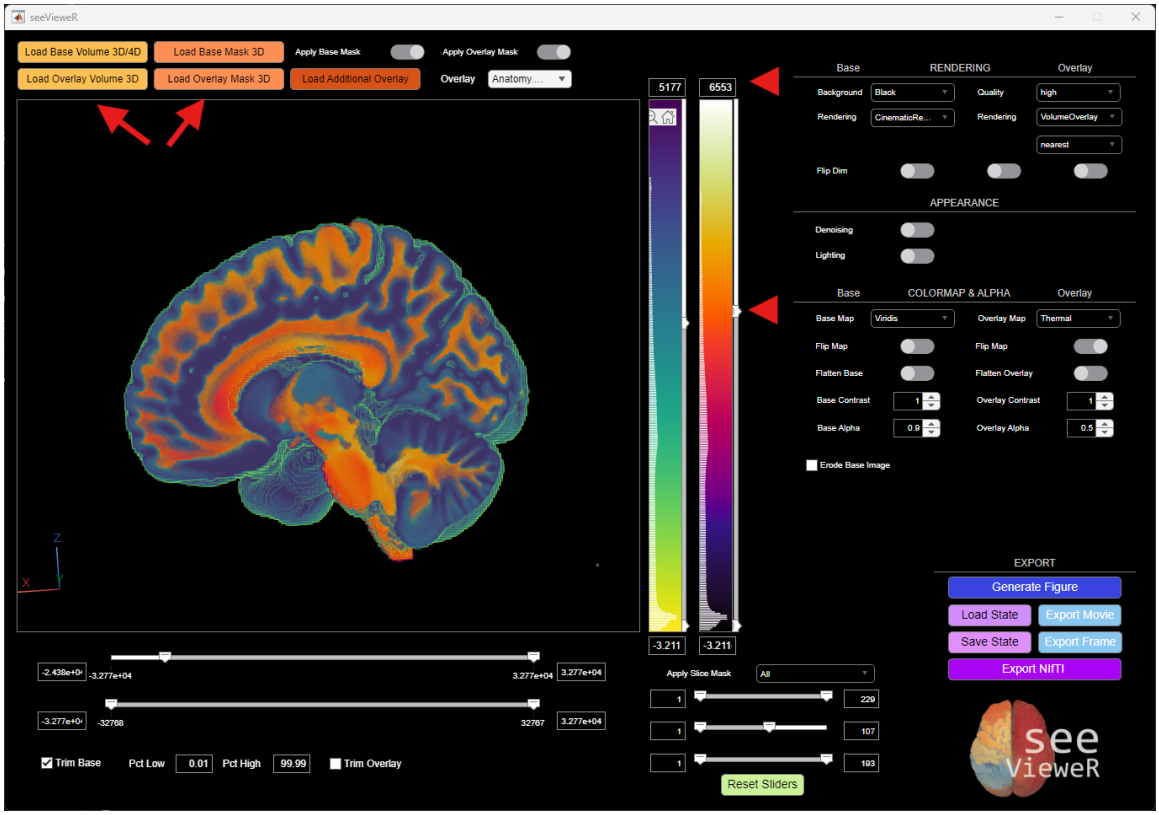
5. Overlay Management

Multiple overlays can be loaded on top of the base image. Each overlay has its own colormap, threshold, and alpha transparency settings. Use the Overlay Intensity Slider to define display limits, and the Threshold Slider to isolate regions of interest. The overlay alpha spinner adjusts the transparency level for blending.

Next lets load an overlay. For now, load the same image as the base image: ‘Anatomy.nii.gz’, and also load a different overlay mask: ‘WM_mask.nii.gz’. Now we can independently apply different masks to the same image (because its loaded as both base and overlay). This way, you can give each region its own colormap.

6. Colormap and Threshold Controls

seeVieweR supports a range of colormaps from MATLAB, Crameri, and ColorBrewer palettes. Use the dropdown menus to change the colormap or flip it using the FlipMap switch. Threshold sliders control which voxels are visible — voxels outside the selected range are masked out in real time. In the example below, Viridis is used for the GM region, while Thermal is used for the WM region. To the right of the respective colorbars, you can also adjust the color scaling, either using the sliders, or by inputting the values directly in the corresponding input boxes.



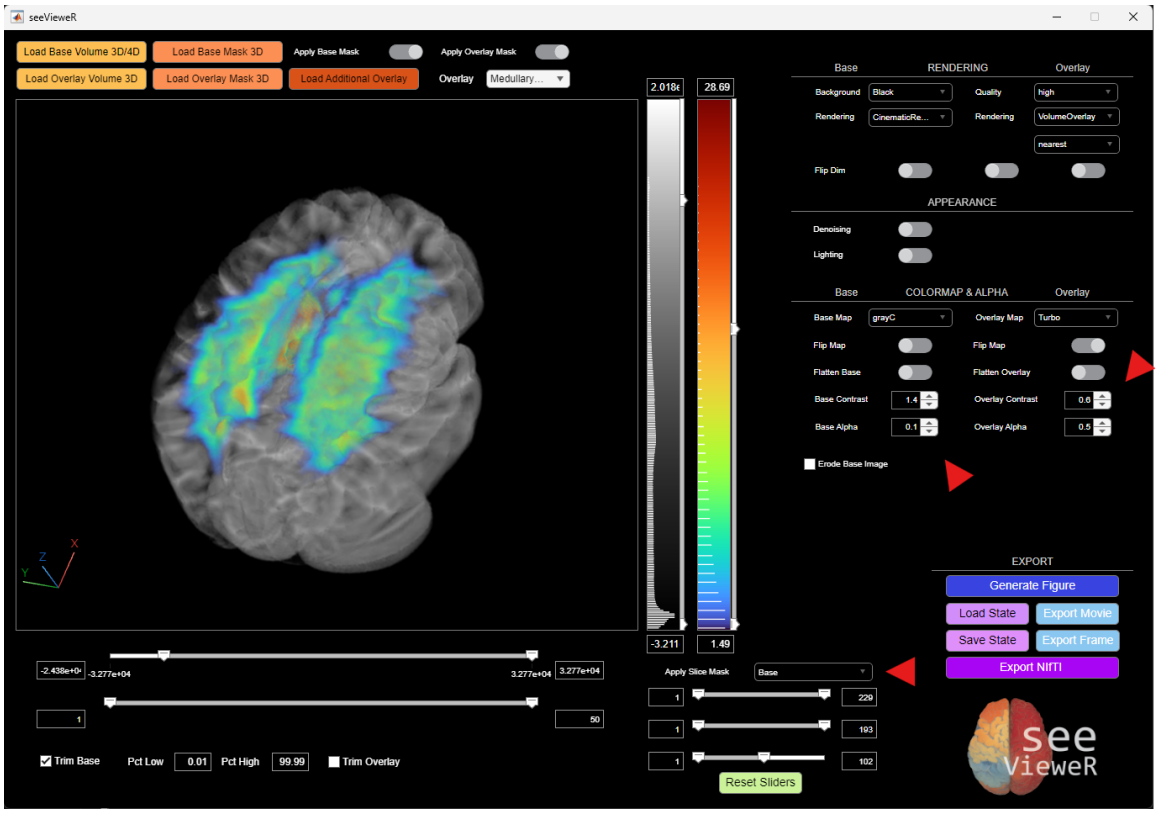
7. Alpha and Contrast Adjustments

The Base Alpha and Overlay Alpha spinners define transparency. For base images, transparency is shaped by the Contrast spinner, which adjusts the gamma curve of the alpha map. Higher contrast values make bright regions more opaque, while lower values enhance soft tissue visibility.

Lets load another overlay image. To do so, press the 'Load Additional Overlay' button. Load the 'Medullary_veins.nii.gz' image. Lets now use grayC for the base image, and Turbo for the overlay. Also set the base image alpha to a very low value (e.g. 0.05). Now we can see the vein atlas through the main brain image. You can play with the slide sliders and slice masking – try to apply the slice mask to only the base (anatomy) image.

We can also adjust the image contrast using the contrast spinners.

Note that we can switch between overlays using the top drop-down:



8. Slice and Mask Controls

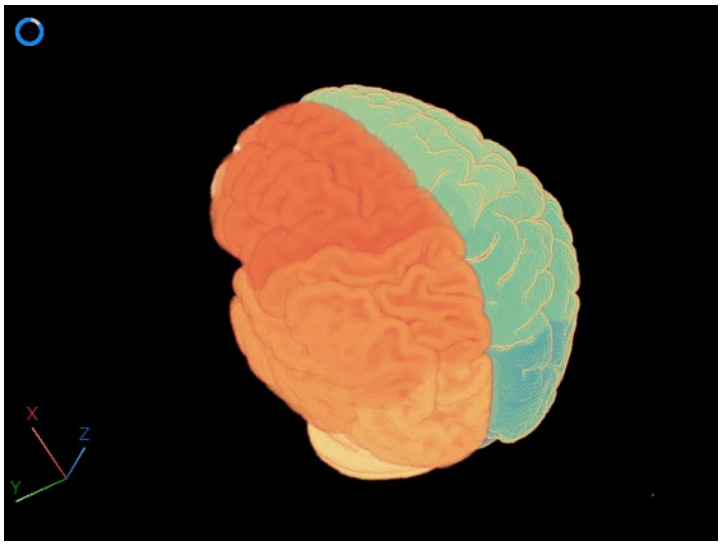
The X/Y/Z slice sliders allow orthogonal slicing through the volume. Masks can be applied to limit visualization to specific regions (e.g., brain or vessel masks). The Apply Mask buttons activate these masks, which can also be used in combination with erosion or skeleton-protection modes for vessel imaging.

9. Exporting Images and Movies

seeVieweR supports high-resolution export of both static frames and movies. It can also generate figures that can form the basis for publications.

- **Export Frame:** saves a PNG of the current 3D view and colorbars.
- **Export Movie:** generates an animated GIF or MP4, optionally with camera rotation.
- **Generate Figure:** saves a PNG/SVG of the base and overlay images.

Let's try! Load the 'AllenBrainAtlas.nii.gz' image as an additional overlay. Try the Spectral colormap and set the base alpha value to zero (removes that image from the display). Then press 'Export Frame' found in the EXPORT panel. You will get a popup where you can select the path to save along with the filename. Press 'Save' and you will have a .png file of what you see in the viewer. The base and overlay color bars will also be saved for use later (e.g. in a figure or presentation).

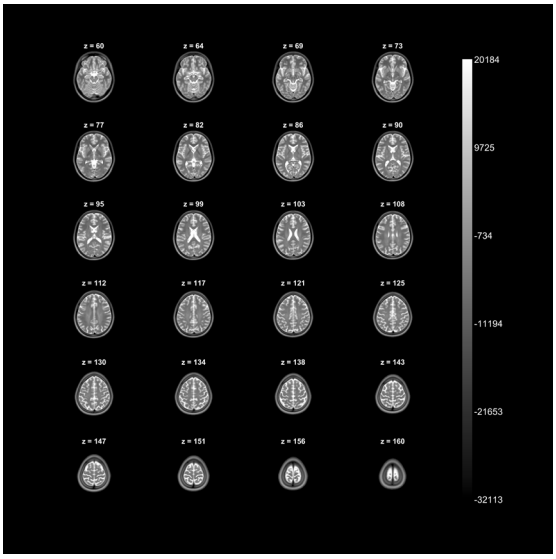


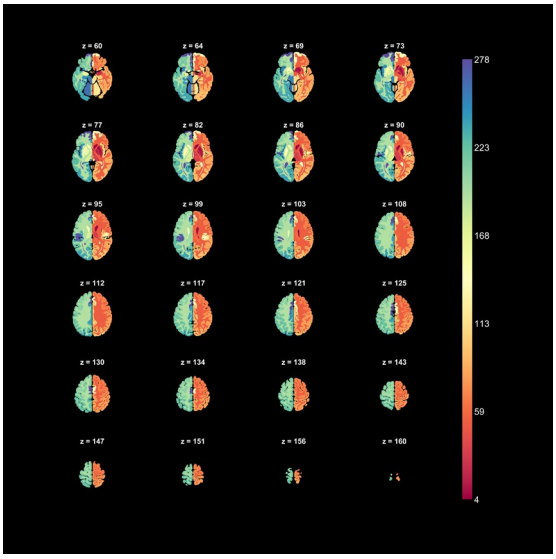
Along with the images, a file called ‘frame_state.mat’ is saved. This is the state file for this export. Later, you can load in the same images and masks and then apply this state file using the ‘Load State’ button. Then you will get back exactly the same view as was used for the figure. State files can also be loaded for other images (e.g. to preserve orientation or scaling between examples) – but this comes with some risks if the data values are not similar between images.

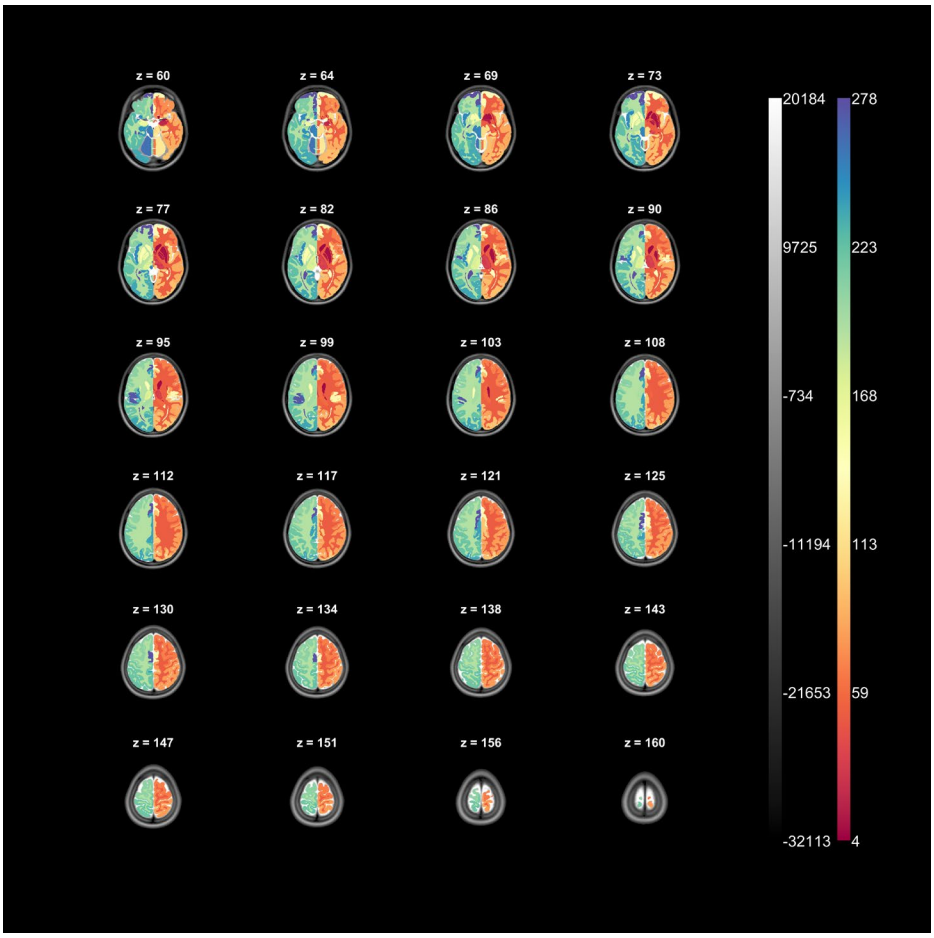
Next let’s try to generate a figure. Press the ‘Generate Figure’ button. You will get the popup here on the left.

- You can select the range of slices you wish to visualize, the scaling (initially taken from the sliders), the orientation and the rotation as well as the colormap (also initialized from the viewer).
- For the overlay, there is also an alpha option (for the fusion plot).
- Overlay fields will copy from base fields but not the other way around. So if you want different rows/columns/slices, then change the overlay after the base options.
- The fusion plot takes the info from the base fields.
- Set the save path and the individual filenames for export in the fields provided
- If masks are switched on in the viewer, the saved images will be masked

Try a slice range between 60-160 … GENERATE!







If you have only a base image loaded, then only one plot is generated. For base and overlay, three plots are generated including the fusion image.

The viewer can also be used to generate video files. To do this, press ‘Export Movie’. If the base image is 4D, then the movie will step through the images one at a time. If the rotation option is selected, a rotation will be generated after stepping through the frames. Note that one the base image can be 4D – this is due to the fact that overlays are interpolated to the grid of the base image and an interpolated overlay can lead to memory explosion.

10. Saving and Loading Viewer States

Use 'Save State' to store the current visualization configuration, including slider values, colormaps, and camera position. Later, 'Load State' restores the viewer to that exact configuration, allowing reproducible visualization setups.

Authored by: Alex A. Bhogal

Version: 1.0
Date: 13-10-2025
Contact: a.bhogal@umcutrecht.nl

www.seeVR.nl