

Resize

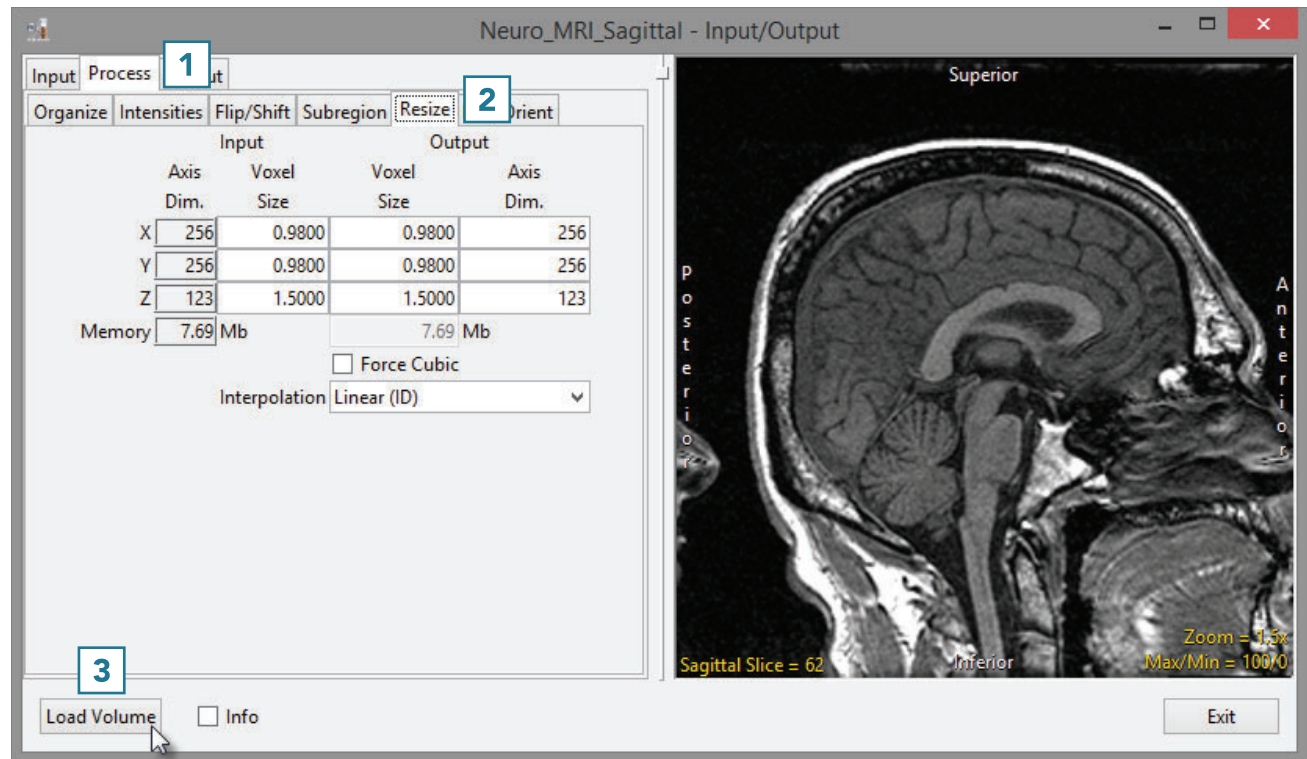
Resize can be used to resample image data. If data is acquired with anisotropic (non-cubic) voxels, resize can be used to resample the data to create an isotropic (cubic) data set.

Resampling Anisotropic Data

- Open Input/Output.
- Navigate to and select the data set to be resampled.
- Select Process [1] and Resize [2].
- Click Load Volume [3] to load the data set without resampling.



Download the Neuro_MRI_Sagittal data set to follow along <http://analyzedirect.com/data/>



Resize (continued)

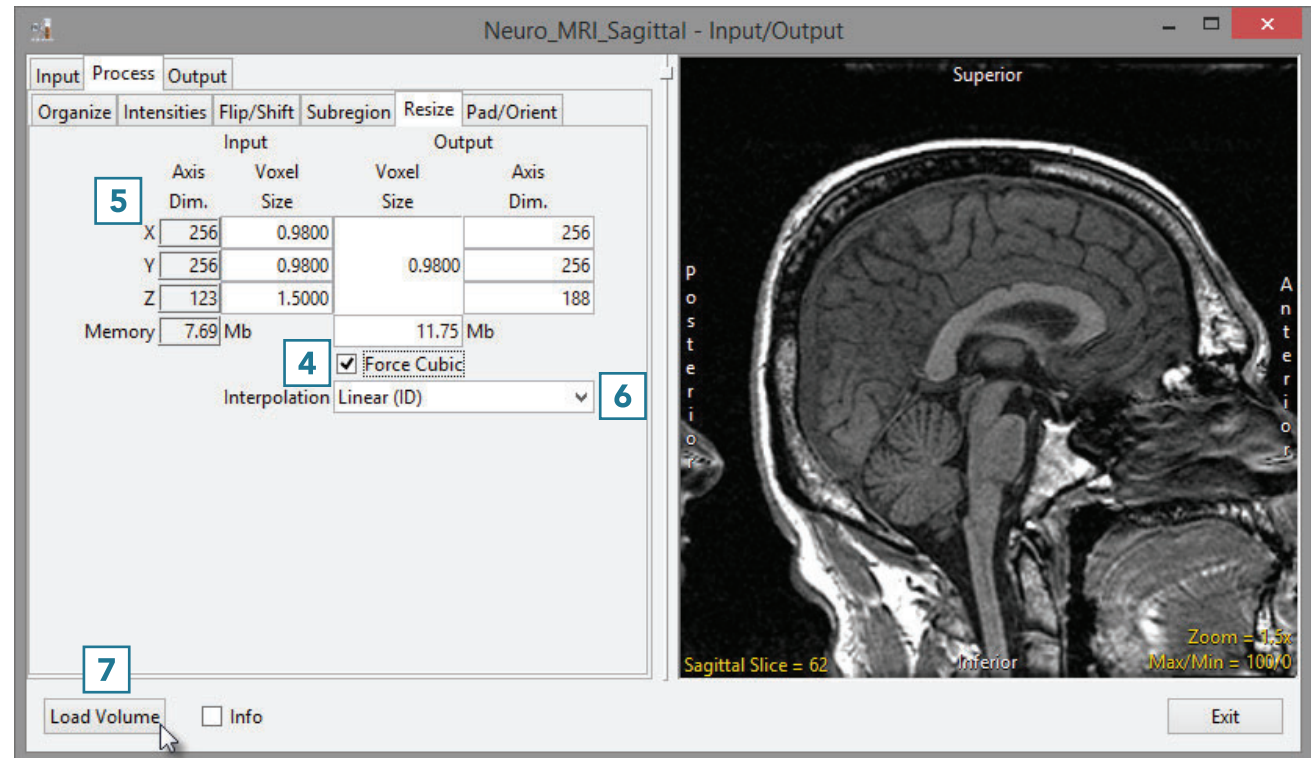
- Check the Force Cubic checkbox [4].

Note the input and output voxel size and the input and output axis dimensions [5]. The data has been resampled at the highest spatial resolution (0.98). The Z dimension which originally had a voxel size of 1.5 has now been resliced to 0.98. To accommodate this, the number of slices in the Z dimension has increased from 123 to 188 ($123 \times 1.5 / 0.98 = 188$).

- The interpolation option used to create the new slice data can be selected from the interpolation drop-down menu [6].

For information about various interpolation types, see Table 3.4 or 6.1.

- Click Load Volume [7] and select Create New. Compare both the original and resampled data sets using Display.





Resize (continued)

Each interpolation type is either input driven (ID) or output driven (OD).

Input driven interpolation algorithms process each voxel in the input volume and map each input voxel into the output space. The actual interpolation takes place in the output space, given the projected contributions of each input voxel into that space. Input driven algorithms will always use every voxel in the input volume as part of the interpolation process.

Output driven algorithms step through each voxel in the output data set and find the corresponding set of voxels in the input data set that map into the output. The interpolation of voxels takes place in the input space, depending on the type of interpolation selected. Output driven algorithms are more common in image processing as they guarantee that every voxel in the output data set will be filled with an interpolated value.

Interpolation Types

Interpolation Type	Description
Nearest Neighbor*	selects the value of the closest voxel to which the interpolation resampling maps
Linear*	applies a linear interpolation of grayscale intensity based on the distance of neighboring voxels from the interpolated voxel
Cubic Spline**	uses a cubic spline function to determine interpolated value
Windowed Sinc**	uses windowed sinc function $\sin(x)/x$ to determine interpolated value
Shape Based	only available for binary data; calculates Euclidean distance, then performs linear interpolation between distance images

*The Nearest Neighbor and Linear interpolation types can be specified as ID or OD.

**The Cubic Spline and Windowed Sinc interpolation types are both OD. Since they use higher-order functions, they account for more neighboring voxels and thus may provide a better estimate for the interpolated value. These algorithms are also more compute-intensive.