BET

<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide>

# **Bet Programs**

* [bet](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide#bet) - the main command-line script which makes calling the core bet2 and betsurf programs easy   
  + [bet2](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide#bet) - the main brain extraction program
  + [betsurf](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide#betsurf) - using a registered T1 and T2 image pair, estimates inner and outer skull surfaces and outer scalp surface
* [Bet](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide#betgui) - the BET GUI

Changes: The old C program called bet has now been discontinued; the bet2 C++ program gives almost identical results. betall has been replaced by the -A and -A2 options in the new bet script. betfunc has been replaced by the -F option in bet. betpremask has been replaced by the -b option in standard\_space\_roi. betsmallz has been replaced by the -Z option in bet.

# **bet and bet2**

By default bet just calls the bet2 program for simple brain extraction. However, it also allows various additional behaviour. The basic usage is:

bet <input> <output> [options]

## **Main bet2 options:**

-o generate brain surface outline overlaid onto original image

-m generate binary brain mask

-s generate rough skull image (not as clean as what betsurf generates)

-n don't generate the default brain image output

-f <f> fractional intensity threshold (0->1); default=0.5; smaller values give larger brain outline estimates

-g <g> vertical gradient in fractional intensity threshold (-1->1); default=0; positive values give larger brain outline at bottom, smaller at top

-r <r> head radius (mm not voxels); initial surface sphere is set to half of this

-c < x y z> centre-of-gravity (voxels not mm) of initial mesh surface.

-t apply thresholding to segmented brain image and mask

-e generates brain surface as mesh in .vtk format.

## **Variations on default bet2 functionality (mutually exclusive options):**

-R This option runs more "robust" brain centre estimation; it repeatedly calls bet2, each time using the same input image and the same main options, except that the -c option (which sets the starting centre of the brain estimation) is set each time to the centre-of-gravity of the previously estimated brain extraction. The primary purpose is to improve the brain extraction when the input data contains a lot of non-brain matter - most likely when there is a lot of neck included in the input image. By iterating in this way the centre-of-gravity should move up each time towards the true centre, resulting in a better final estimate. The iterations stop when the centre-of-gravity stops moving, up to a maximum of 10 iterations.

-S This attempts to cleanup residual eye and optic nerve voxels which bet2 can sometimes leave behind. This can be useful when running SIENA or SIENAX, for example. Various stages involving standard-space masking, morphpological operations and thresholdings are combined to produce a result which can often give better results than just running bet2.

-B This attempts to reduce image bias, and residual neck voxels. This can be useful when running SIENA or SIENAX, for example. Various stages involving FAST segmentation-based bias field removal and standard-space masking are combined to produce a result which can often give better results than just running bet2.

-Z This can improve the brain extraction if only a few slices are present in the data (i.e., a small field of view in the Z direction). This is achieved by padding the end slices in both directions, copying the end slices several times, running bet2 and then removing the added slices.

-F This option uses bet2 to determine a brain mask on the basis of the first volume in a 4D data set, and applies this to the whole data set. This is principally intended for use on FMRI data, for example to remove eyeballs. Because it is normally important (in this application) that masking be liberal (ie that there be little risk of cutting out valid brain voxels) the -f threshold is reduced to 0.3, and also the brain mask is "dilated" slightly before being used. -A This runs both bet2 and betsurf programs in order to get the additional skull and scalp surfaces created by betsurf. This involves registering to standard space in order to allow betsurf` to find the standard space masks it needs.

-A2 <T2> This is the same as -A except that a T2 image is also input, to further improve the estimated skull and scalp surfaces. As well as carrying out the standard space registration this also registers the T2 to the T1 input image.

The old script betpremask has been replaced with the -b option in standard\_space\_roi. This uses flirt to register the input image to a standard space whole-head image; the resulting transform is then inverted and a standard-space brain mask is transformed into the space of the input image, and then applied to this to create the output.

# **Bet GUI**

## **Main Options**

First set the **Input image** filename and the **Output image** filename. The output name is also used to generate the filenames for any optional outputs that are turned on in **Advanced options**.

Changing **Fractional intensity threshold** from its default value of 0.5 will cause the overall segmented brain to become larger (<0.5) or smaller (>0.5). This threshold must lie between 0 and 1.

If you change the options menu from the default setting of **Run standard brain extraction using bet2** then you will get one of the **Variations on default bet2 functionality** described above.

## **Advanced Options**

By default the only output from BET is an image with all non-brain matter removed - this is the **Output brain-extracted image option**.

The **Output binary brain mask image** option tells BET to output a binary brain mask (0 outside of the brain and 1 inside). This can then be used in later processing, to mask other images derived from the original.

The **Apply thresholding to brain and mask image** option causes BET to apply thresholding to the segmented brain image (and also the brain mask if selected). Thus, inside the estimated brain, some voxels can be "zeroed" after segmentation, if their intensity falls below an automatically estimated threshold.

The **Output exterior skull surface image** option tells BET to produce an image which is an estimate of the exterior surface of the skull. All non-skull-surface points are 0 in this image, and skull-surface points are 100.

The **Output brain surface overlaid onto original image** option creates an output image which is a copy of the original image, with additionally an outline of the brain's surface drawn on top.

Changing **Threshold gradient** from its default value of 0 causes a gradient to be apply to the previous threshold. Thus setting a positive value here causes the primary threshold to be reduced at the bottom of the brain, giving a larger brain estimate there, and a smaller estimate of the brain towards the top of the image. This value must lie between -1 and 1.

# **betsurf**

betsurf is the program that produces the three additional surfaces (inner & outer skull, outer scalp). It can output these surfaces as filled-in binary mask images, surface-only binary mask images and mesh format files.

betsurf requires ideally good resolution T1- and T2-weighted input images. It can work only from just a T1 image, but the results will generally not be as good. The images should ideally be better than 2x2x2mm - the smaller the voxel size the better. betsurf requires that bet2 has already been run on the T1 image, with the -e mesh output option for the brain surface. It also requires that the T2 image already be aligned (e.g., using [flirt](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT)) to the T1. Finally, it requires that you supply a transformation from the T1 to standard space (e.g., generated by flirt) as it uses standard-space masks in order to constraint some of the mesh fitting.

The .vtk mesh files contain mm co-ordinates. If the image has radiological ordering ( see fslorient ) then the mm co-ordinates are the voxel co-ordinates scaled by the mm voxel sizes. If the image has neurological ordering the y and z co-ordinates are the same as for radiological ordering, but the x co-ordinates will be inverted ( x\_mm = ( n - 1 - x ) \* x\_dim ) where n is the number of voxels in the x-direction, x is the voxel co-ordinate and x\_dim is the voxel size in mm.

As an example of how to use bet2 and betsurf together, see the -A and -A2 options in the bet script. The -A2 option:

Registers the T2 to the T1 image. Registers the T1 image to standard space (not to transform the image but just to get the transform matrix itself). Runs bet2 to get the initial brain extraction from the T1 image. Runs betsurf to get the other head surfaces.