1 Program 3dmerge

1.1 Purpose

Program 3dmerge is for editing and/or merging of 3D datasets. The editing options, which are applied to each individual 3D dataset, include:

- Discard negative intensities, or take their absolute values.
- Discard intensities below a given 'clipping' level.
- Use the threshold in a fith (or fico, or fitt, or fift) file to censor the dataset.
- Blur voxels with a Gaussian tapering function.
- Discard voxels not in a cluster of at least a given volume.
- Apply a spatial filter to voxel intensities.
- Linearly scale voxel intensities so that the largest present is set to 10000.

These functions are applied in the order given.

Various options are provided for forming a single 3D dataset from a group of 3D datasets (the merging options). The merging functions include:

- Discard voxels that aren't nonzero in a given number of datasets (principally used to edit out one-time-only results).
- Combine using voxel means or voxel max.
- After combination, again discard voxels not in a cluster of at least a given volume.

These functions are applied in the order given.

For some applications, you may wish to edit functional datasets using 3dmerge before using 3dproject or 3dclust on them.

1.2 Usage

The command line format for program 3dmerge is as follows:

3dmerge [options] datasets . . .
where the options are:

1.3 Editing Options Applied to Each Input Dataset

-1thtoin Copy threshold data over intensity data. This is only valid for datasets with some thresholding statistic attached. All subsequent operations apply to this substituted data.

-2thtoin The same as -1thtoin, but do NOT scale the threshold values from shorts to floats when processing. This option is only provided for compatibility with the earlier versions of the AFNI package '3d*' programs.

-1noneg Zero out voxels with negative intensities.

-1abs Take absolute values of intensities.

-1clip val Clip intensities in range [-val,val] to zero.

-2clip v1 v2 Clip intensities in range [v1,v2] to zero.

-1uclip val , -2uclip v1 v2
These options are like the above, but do not apply any automatic scaling factor that may be attached to the data. These are for use only in special circumstances. (The 'u' means 'unscaled'. Program '3dinfo' can be used to find the scaling factors.) N.B.: Only one of these 'clip' options can be used; you cannot combine them to have multiple clipping executed.

-1thresh thr Use the threshold data to censor the intensities (only valid for 'fith', 'fico', or 'fitt' datasets). N.B.: The value 'thr' is floating point, in the range 0.0<thr<1.0 for 'fith' and 'fico' datasets, and 0.0<thr<32.7 for 'fitt' datasets.

Gaussian blur options:

-1blur_sigma bmm Isotropic Gaussian blur with sigma = bmm (in mm).

-1blur_rms bmm Isotropic Gaussian blur with rms deviation = bmm.

-1blur_fwhm bmm Isotropic Gaussian blur with FWHM = bmm.

-t1blur_sigma bmm Isotropic Gaussian blur of the threshold values with sigma = bmm (in mm).

-t1blur_rms bmm Isotropic Gaussian blur of the threshold values with rms deviation = bmm.

-t1blur_fwhm bmm Isotropic Gaussian blur of the threshold values with FWHM = bmm.
N.B.: The 3 '-1blur' and 3 '-t1blur' options just provide different ways of specifying the radius used for the blurring function. The relationships among these specifications are

$$\text{sigma} = 0.57735027 * \text{rms} = 0.42466090 * \text{fwhm}$$

The requisite convolutions are done using FFTs; this is one of the slowest operations among the editing options.

-1zvol x1 x2 y1 y2 z1 z2 Zero out entries inside the 3D volume defined by x1≤x≤x2, y1≤y≤y2, z1≤z≤z2. N.B.: The ranges of x, y, z in a dataset can be found using the '3dinfo' program. Dimensions are in mm.

N.B.: This option may not work correctly at this time, but I've not figured out why!

The following cluster options are mutually exclusive:

-1clust rmm vmul Form clusters with connection distance rmm and clip off data not in clusters of volume at least vmul microliters.

-1clust_mean rmm vmul Same as -1clust, but all voxel intensities within a cluster are replaced by the average intensity of the cluster.

-1clust_max rmm vmul Same as -1clust, but all voxel intensities within a cluster are replaced by the maximum intensity of the cluster.

-1clust_amax rmm vmul Same as -1clust, but all voxel intensities within a cluster are replaced by the maximum absolute value of the intensity of the cluster.

-1clust_smax rmm vmul Same as -1clust, but all voxel intensities within a cluster are replaced by the maximum signed absolute value of the intensity of the cluster, i.e., the sign is preserved.

-1clust_size rmm vmul Same as -1clust, but all voxel intensities within a cluster are replaced by the size of the cluster (in multiples of vmul).

-1clust_order rmm vmul Same as -1clust, but all voxel intensities within a cluster are replaced by the cluster size index (largest cluster = 1, next largest = 2, etc.).

The following commands produce erosion and dilation of 3D clusters. These commands assume that one of the -1clust commands has been used. The purpose is to avoid forming strange clusters with 2 (or more) main bodies connected by thin 'necks'. Erosion can cut off the neck. Dilation will minimize erosion of the main bodies.

Note: Manipulation of values inside a cluster (-1clust commands) occurs AFTER the following two commands have been executed.

-1erode pv For each voxel, set the intensity to zero unless *pv* % of the voxels within radius rmm are nonzero.

-1dilate Restore voxels that were removed by the previous command if there remains a nonzero voxel within rmm.

The following spatial filtering options are mutually exclusive:

-1filter_mean rmm Set each voxel to the average intensity of the voxels within a radius of rmm.

-1filter_nzmean rmm Set each voxel to the average intensity of the non-zero voxels within a radius of rmm.

-1filter_max rmm Set each voxel to the maximum intensity of the voxels within a radius of rmm.

-1filter_amax rmm Set each voxel to the maximum absolute value of the intensity of the voxels within a radius of rmm.

-1filter_smax rmm Set each voxel to the maximum absolute value (preserving the sign) of the intensity of the voxels within a radius of rmm.

-1filter_aver rmm Same idea as '_mean', but implemented using a new code that should be faster.

The following threshold spatial filtering options are mutually exclusive:

-t1filter_mean rmm Set each correlation or threshold voxel to the average of the voxels within a radius of rmm.

-t1filter_nzmean rmm Set each correlation or threshold voxel to the average of the non-zero voxels within a radius of rmm.

-t1filter_max rmm Set each correlation or threshold voxel to the maximum of the voxels within a radius of rmm.

-t1filter_amax rmm Set each correlation or threshold voxel to the maximum absolute value of the intensity of the voxels within a radius of rmm.

-t1filter_smax rmm Set each correlation or threshold voxel to the maximum absolute value (preserving the sign) of the intensity of the voxels within a radius of rmm.

-t1filter_aver rmm Same idea as '_mean', but implemented using a new code that should be faster.

-1mult factor Multiply intensities by the given factor.

-1zscore If the sub-brick is labeled as a statistic from a known distribution, it will be converted to an equivalent $N(0,1)$ deviate (or a 'z score'). If the sub-brick is not so labeled, nothing will be done.

The above '-1' options are carried out in the order given above, regardless of the order in which they are entered on the command line.

-datum type Coerce the output data to be stored as the given type, which may be byte, short, or float.
N.B.: Byte data cannot be negative. If this datum type is chosen, any negative values in the edited and/or merged dataset will be set to zero.

-keepthr When using 3dmerge to edit exactly one dataset of a functional type with a threshold statistic attached, normally the resulting dataset is of the 'fim' (intensity only) type. This option tells 3dmerge to copy the threshold data (unedited in any way) into the output dataset.
N.B.: This option is ignored if 3dmerge is being used to combine 2 or more datasets.
N.B.: The -datum option has no effect on the storage of the threshold data. Instead use '-thdatum type'.

-doall Apply editing and merging options to ALL sub-bricks uniformly in a dataset.
N.B.: All datasets must have the same number of sub-bricks when using the -doall option.
N.B.: The threshold specific options (such as -1thresh, -keepthr, -trfisher, etc.) are not compatible with the -doall command. Neither are the -1dindex or the -1tindex options, described below.
N.B.: All labels and statistical parameters for individual sub-bricks are copied from the first dataset. It is the responsibility of the user to verify that these are appropriate. Note that sub-brick auxiliary data can be modified using program 3drefit.

-1dindex j Use sub-brick #j as the data source.

-1tindex k Use sub-brick #k as the threshold source.

With these commands, you can operate on any given sub-brick of the input dataset(s) to produce as output a 1 brick dataset. If desired, a collection of 1 brick datasets can later be assembled into a multi-brick bucket dataset using program 3dbucket or into a 3D+time dataset using program 3dTcat.

N.B.: If these options aren't used, j=0 and k=1 are the defaults.

1.4 Merging Options Applied to Form the Output Dataset

Here we describe different ways to combine several datasets. The following '-g' options are mutually exclusive!

-gmean Combine datasets by averaging intensities (including zeros) – this is the default.

-gnzmean Combine datasets by averaging intensities (not counting zeros).

-gmax Combine datasets by taking max intensity (e.g., -7 and 2 combine to 2)

-gamax Combine datasets by taking max absolute intensity (e.g., -7 and 2 combine to 7)

-gsmax Combine datasets by taking max signed intensity (e.g., -7 and 2 combine to -7)

-gcount Combine datasets by counting number of 'hits' in each voxel (a 'hit' occurs if a voxel is nonzero).

-gorder Combine datasets in order of input:

- If a voxel is nonzero in dataset #1, then that value goes into the voxel.
- If a voxel is zero in dataset #1 but nonzero in dataset #2, then the value from #2 is used.
- And so forth: the first dataset with a nonzero entry in a given voxel 'wins'

-gfisher Take the arctanh of each input, average these, and output the tanh of the average. If the input datum is 'short', then input values are scaled by 0.0001 and output values by 10000. This option is for merging bricks of correlation coefficients.

1.5 Merging Operations Applied to the Threshold Data

Here we describe different ways to combine the thresholds. If none of these are given, the thresholds will not be merged and the output dataset will not have threshold data attached. Note that the following '-tg' command line options are mutually exclusive, but are independent of the '-g' options given above for merging the intensity data values.

-tgfisher This option is only applicable if each input dataset is of the 'fico' or 'fith' types — functional intensity plus correlation or plus threshold. (In the latter case, the threshold values are interpreted as correlation coefficients.) The correlation coefficients are averaged as described by -gfisher above, and the output dataset will be of the fico type if all inputs are fico type; otherwise, the output datasets will be of the fith type. N.B.: The difference between the -tgfisher and -gfisher methods is that -tgfisher applies to the threshold data stored with a dataset, while -gfisher applies to the intensity data. Thus, -gfisher would normally be applied to a dataset created from correlation coefficients directly, or from the application of the -1thtoin option to a fico or fith dataset.

1.6 Optional Ways to Postprocess the Combined Results

These options may be combined with the above methods. Any combination of these options may be used.

-ghits count Delete voxels that aren't $\neq 0$ in at least count datasets ($\neq 0$ is a 'hit')

-gclust rmm vmul Form clusters with connection distance rmm and clip off data not in clusters of volume at least vmul microliters

The '-g' and '-tg' options apply to the entire group of input datasets.

1.7 Options that Control the Names of the Output Dataset

-session dirname Write output into given directory (default=./)

-prefix pname Use 'pname' for the output directory prefix (default=mrg)

1.8 Notes

- If only one dataset is read into this program, then the '-g' options do not apply, and the output dataset is simply the '-1' options applied to the input dataset (i.e., edited).
- A merged output dataset is ALWAYS of the intensity only variety.
- Complex-valued datasets cannot be merged.
- This program cannot handle time-dependent datasets.
- Note that the input datasets are specified by their .HEAD files, but that their .BRIK files must exist also!

1.9 Examples

Example 1. An investigator wishes to combine functional images fred.01+tlrc, fred.02+tlrc, fred.03+tlrc, and fred.04+tlrc (.BRIK and .HEAD) by taking the average of the intensity, at each voxel, over the 4 datasets. A batch command file to accomplish this is as follows:

```
3dmerge -gmean -prefix fred.gmean \
fred.01+tlrc fred.02+tlrc fred.03+tlrc fred.04+tlrc
```

After execution, file fred.gmean+tlrc (.BRIK and .HEAD) will contain the average intensities of the input data sets.

Example 2. The investigator decides, after examining the previous output using program afni, that it would be better to combine, by averaging, only those voxels which show a significant functional activation. This is accomplished through use of the -1thresh command. Assuming that the functional images fred.01+tlrc through fred.04+tlrc are of the afni 'fico' type, i.e., functional intensity + correlation coefficient, the correlation coefficient sub-brick can be used to set a threshold to censor the intensities. A batch command file to do this might be:

```
3dmerge -gmean –1thresh 0.5 -prefix fred.thresh.gmean \
fred.01+tlrc fred.02+tlrc fred.03+tlrc fred.04+tlrc
```

Only voxels whose correlation coefficient is greater than 0.5 (in absolute value) will be averaged, and the output is sent to file fred.thresh.gmean+tlrc (.BRIK and .HEAD).

Example 3. After examining the previous output, the investigator decides that slight mis-alignment of the functional images has occurred. Therefore, voxels that should overlap in the different images do not. One way to cope with this is to artificially enlarge the regions of functional activation. This can be accomplished in several different ways. One possibility is to use a Gaussian blur, as in:

```
 3dmerge -gmean –1thresh 0.5 -1blur_sigma 3 \
-prefix fred.thresh.blur.gmean \
 fred.01+tlrc fred.02+tlrc fred.03+tlrc fred.04+tlrc
```

Each input functional image is "blurred" by convolution with a Gaussian function with $\sigma = 3$ mm. Another possibility is to use one of the '-1filter' commands, such as:

```
 3dmerge -gmean –1thresh 0.5 -1filter_nzmean 2 \
-prefix fred.thresh.filtnzmean.gmean \
 fred.01+tlrc fred.02+tlrc fred.03+tlrc fred.04+tlrc
```

The result of the '-1filter_nzmean 2' command is as follows: the intensity of each voxel in each input data set is replaced by the average of the voxel intensities of all voxels (not counting voxels having zero intensity) within a radius of 2 mm of the original voxel. This has the effect of “spreading” the regions of non-zero activation.

Example 4. The above example showed how to “grow” regions of functional activation. However, this also “grows” isolated activated voxels, which may be due to noise. Since clusters of activated voxels are far less likely to be the result of random noise than isolated voxels, the researcher decides to keep only clusters of activated voxels from the input data sets. This is accomplished using one of the ‘-1clust’ commands. For example, consider the following batch command file:

```
 3dmerge -gmean –1thresh 0.5 –1clust_mean 2 300 -1filter_nzmean 2 \
-prefix fred.thresh.clustmean.filtnzmean.gmean \
 fred.01+tlrc fred.02+tlrc fred.03+tlrc fred.04+tlrc
```

The ‘-1clust_mean 2 300’ does the following: For each input data set, the program identifies clusters of active (nonzero) voxels. Two activated voxels are in the same cluster if they are separated by not more than 2 mm. Therefore, every voxel is a member of exactly 1 cluster (even if the cluster contains only 1 voxel). After the clusters have been identified, any cluster with volume less than 300 mm^3 is discarded. Therefore, only active voxels belonging to a cluster of volume $\geq$ 300 mm^3 remain. Now, since the ‘-1clust_*mean*’ option is used, the intensity of each voxel within a cluster is replaced by the average intensity of the voxels within that cluster. Since this is followed by the -1filter_nzmean, the effect is to “grow” those clusters having the specified minimum size. The output is stored in file fred.thresh.clustmean.filtnzmean.gmean+tlrc (.HEAD and .BRIK).

Example 5. An experimenter has created a set of 3D functional images. These images were generated for a group of 10 subjects. Each subject was subjected to 4 different tests on 5 consecutive days, for a total of 200 functional volumes. Due to the amount of computer time required to process so many images, the experimenter decided that it would be best to “pre-process” the images, producing data files that contain the clustered and filtered images.

Suppose that the data file names have the format: fred.ii.jj.kk+orig (.HEAD and .BRIK), where ii = subject id number, jj = test number, and kk = day number. A C-shell batch command file to threshold, cluster, and filter each of the data sets is given by:

```
foreach subject ( 1 2 3 4 5 6 7 8 9 10 )
  foreach test ( 1 2 3 4 )
    foreach day ( 1 2 3 4 5 )
      3dmerge –1thresh 0.5 –1clust_mean 2 300 -1filter_nzmean 2 \
     -prefix fred.proc.${subject}.${test}.${day} \
      fred.${subject}.${test}.${day}+orig
    end
  end
end
```

Each input file fred.ii.jj.kk+orig (.HEAD and .BRIK) produces a corresponding output file fred.proc.ii.jj.kk+orig (.HEAD and .BRIK). The user may now experiment with different ways of combining these output files, without having to repeat the lengthy process of thresholding, clustering, and filtering for each individual data set.