



Image calculations and transformations (using SPM)
This software version: **09-Nov-2017**
(C) Robert J Ellis (<http://tools.robjellis.net>)

Documentation for **imcalc** (SPM 5/8/12)

Robert J Ellis | <http://tools.robjellis.net>

Updates

- November 2017
 - o Expanded documentation.
 - o Batch processing of cluster images (option [17]) is now possible. (Thanks, [Patrick McConnell](#), for the suggestion).
 - o Confirmed compatibility with SPM12 (version: 20 Oct 2016)
 - o Confirmed automatic image reslicing works in SPM{5/8/12}.
 - o Written image files are now written to the smallest file size (i.e., data type) available.
- March 2013
 - o Checked compatibility with SPM12 beta
 - o A warning error (only occurred in SPM5) regarding Analyze orientation and image masks has been deprecated.
- November 2012
 - o Option [0] now enables user-entered expressions on a 4D volume (equivalent to the "matrix X" option in SPM's native imcalc)
- September 2012
 - o no functions require the MATLAB **stats toolbox** (alternative functions are packaged with this release)

Installation

- Unzip *imcalc.zip* into your target directory (preferred: *matlab/spm{5/8/12}/toolboxes/imcalc*)
- Ensure that this folder is on the Matlab path using `pathtool`
- It is **highly recommended** that you also have an image viewer on hand (e.g., [MRICron](#)) to view the resultant images and confirm they are what you expected.

Version compatibility

- `imcalc` is designed to work under the following SPM / Matlab pairings:
 - SPM5 (rev. 1807, compatible with Matlab R13SP1 to R2006b);
 - SPM8 (rev. 6313, compatible with Matlab R14SP3 to R2015a); and
 - SPM12 (rev. 6906, compatible with Matlab R2007a to R2017b)
- However, if any of the following errors appear in conjunction with using SPM5 or SPM8 on a more recent version of Matlab, it is recommended that you upgrade to SPM12:
 - No method 'delete' with matching signature found for class 'matlab.ui.control.UIControl'.
- Also, if you alternate between different versions of SPM on the same machine, you may need to type `clear classes` (as prompted by Matlab) before launching `imcalc`

Running the program

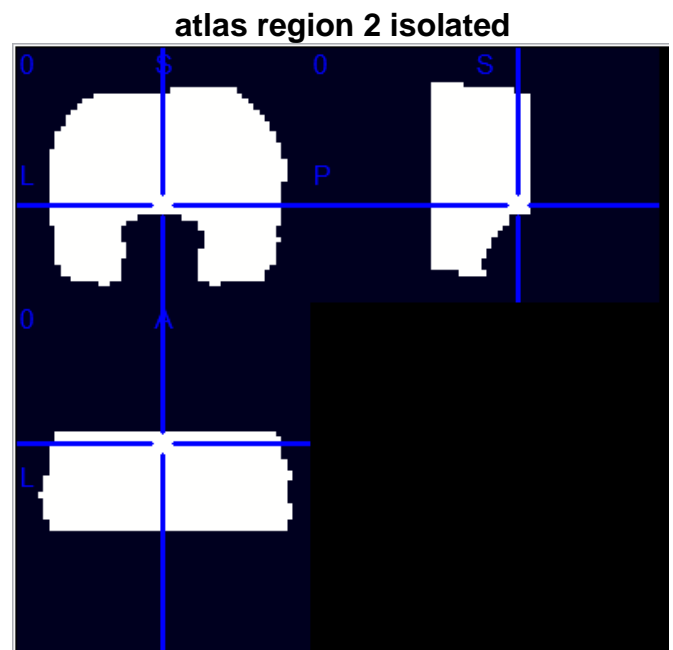
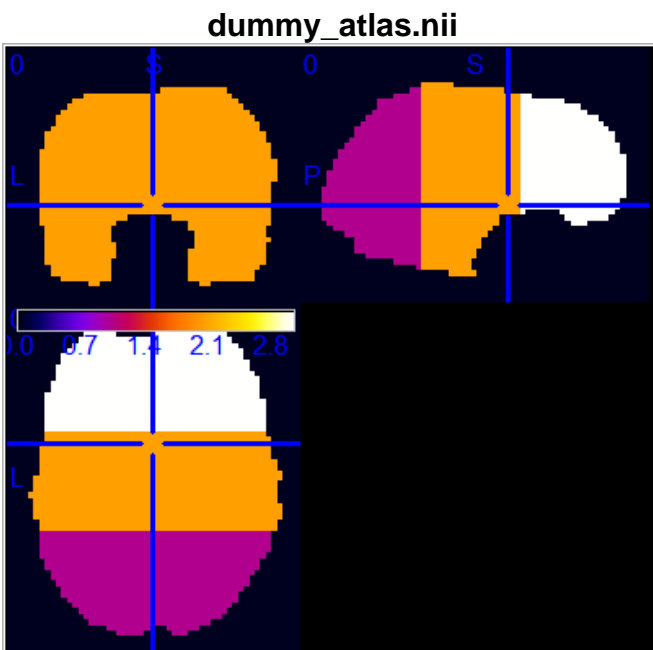
- at the `>>`, type `imcalc`
 - **Hint:** if you want to use option X, you can specify this directly: `imcalc(X)`
 - Example: to run option [7], type `imcalc(7)`
 - **Note #1:** Launching the program will initiate an automatic check to make sure that SPM is on the Matlab path, and will prompt for user to identify the main SPM directory if not. `imcalc` will be added to the top of the Matlab path, which is essential for proper functioning.
 - **Note #2:** For all steps, the user is able to select an explicit mask (`.img` or `.nii`) to mask all images to a common template
 - **Note #3:** Example images in this documentation are displayed using [MRIcron](#)
 - **Note #4:** The “sample_data” folder contains images used to illustrate the features of vis:
 - two first-level `con*.nii` images (`con_0006` and `con_0007`) from the SPM “face_rfx” data set
 - a “dummy_atlas.nii” image, created using MarsBaR, in which the `face_rfx` brain template is masked into three broad regions.

- Options are listed below. Detailed explanations are provided on the following pages.

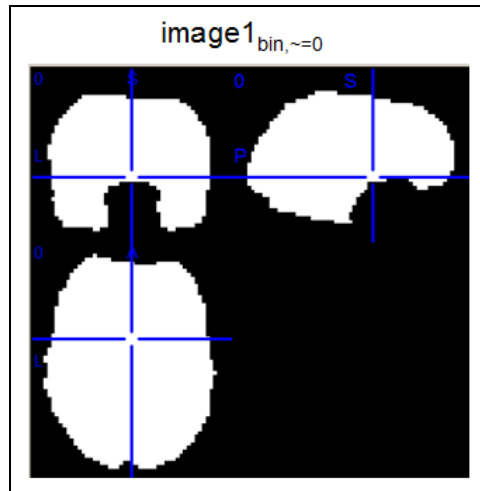
```
[0] user entered expression (either group-wise or image-wise)
[1] binarize non-zero voxels
[2] binarize/threshold each image
[3] binarize/threshold each image, then sum
[4] [binarize non-zero voxels] then sum
[5] binarize non-zero voxels, sum, rebinarize
[6] voxelwise calculations on pairs
[7] flip sign of all non-zero voxels
[8] x-flip image along y = 0
[9] apply an inclusive/exclusive mask
[10] T-to-Z transform
[11] Winsorize (cap) extreme values
[12] Z-score transform of image relative to its global mean and SD
[13] percap: cap at a fixed percentile (upper and lower)
[14] divide by X times the SD of non-zero voxels
[15] write single voxels to a .nii
[16] create a cluster image
[17] split cluster image(s) into constituent images
[18] linear XYZ translation
[20] write hemisphere masks from template
[21] homotopic voxel calculations
[22] replace zeros/NaNs with ___
[23] inter-subject proportion maps
[24] pad image with extra voxels
```

Options [0] to [11] in detail:

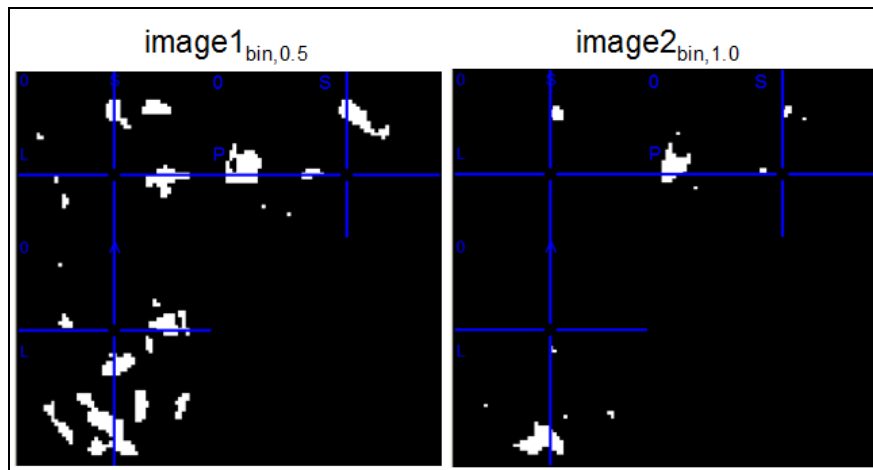
- [0] user-entered expression to perform on volume(s)
 - **Note:** This feature is more advanced than options [1] to [11], but is listed first for historical reasons
 - **Example use case:** Isolating a particular ROI from a brain atlas. Below is a dummy atlas with three regions, labeled 1, 2, and 3 (see image below):
- In imcalc:
 - Specify whether one or two groups (i.e., pairwise image operations)
 - One group is default
 - Select the to-be-transformed image(s)
 - *sample_data* → *dummy_atlas.nii*
 - Replace zero-valued voxels?
 - Yes by default
 - Enter the desired expression; don't need to put it in quotes
 - To isolate the "central" ROI (voxel values = 2) enter `v == 2`
 - Enter a suffix to write to the image
 - How about `roi2`
 - Apply an additional mask?
 - "No" is default



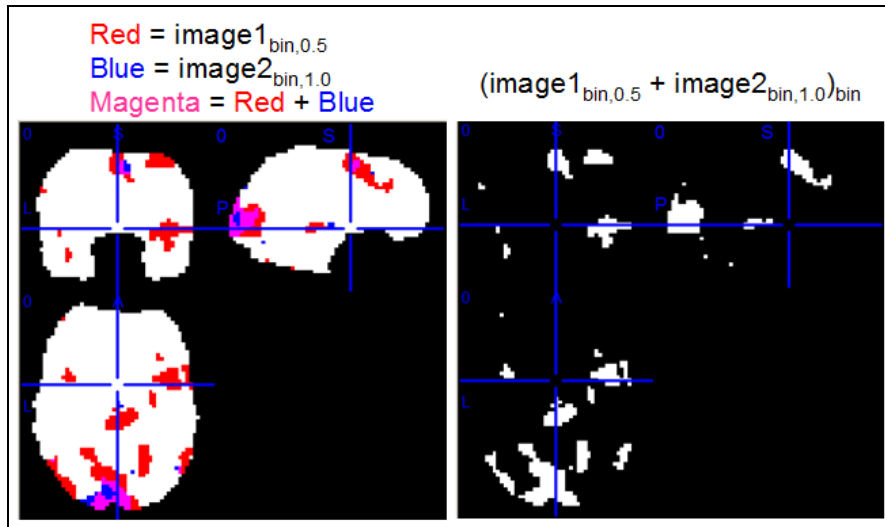
- **[1] binarize non-zero voxels**
 - Select the N images
 - Binarize all non-zero voxels to 1.0



- **[2] binarize each image to a unique threshold;**
 - Select the N images
 - Enter an $[N \times 1]$ or $[1 \times N]$ vector of values, giving the unique threshold at which each image should be binarized

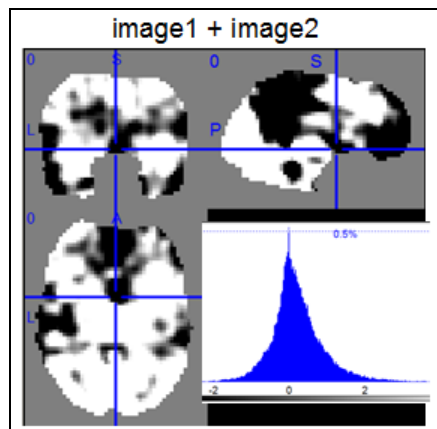


- **[3] binarize each image to a unique threshold, then sum;**
 - As in [2], but then sums the binarized images into a single image (with values 0 to N)

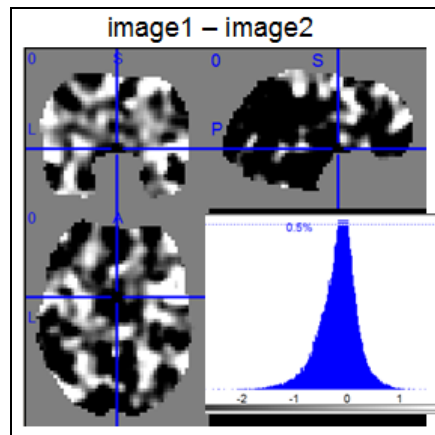


- **[4] binarize non-zero voxels, then sum;**
 - Select the N images
 - Enter an $[N \times 1]$ or $[1 \times N]$ vector of values, giving the unique threshold at which each image should be binarized
 - imcalc then sums the binarized images into a single image (with values 0 to N)
- **[5] binarize non-zero voxels, sum, rebinarize;**
 - Select the N images
 - Binarize all non-zero voxels to 1
 - imcalc then sums these binarized images, and rebinarizes all voxels > 0 to 1
- **[6] pairwise operation (add sub mult div ... etc.);**
 - Select N "group 1" images
 - Select N "group 2" images
 - Operation:

1. sum: $G1 + G2$

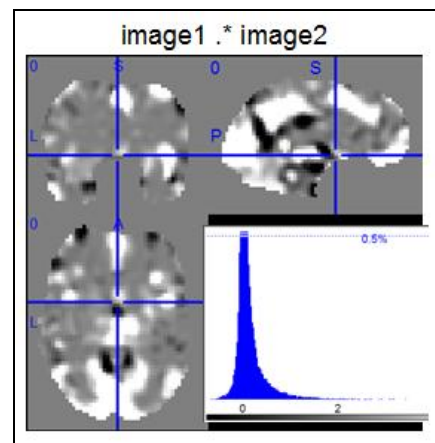


2. subtract: $G1 - G2$



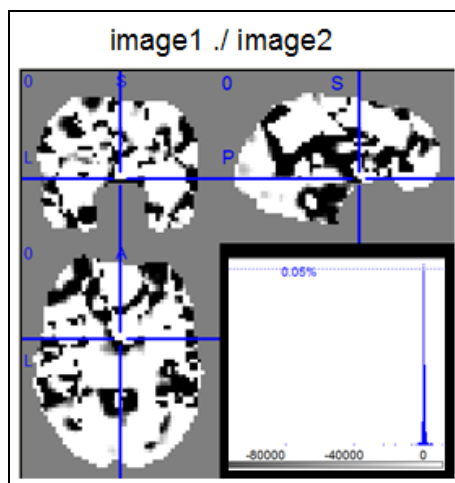
3. multiply: $G1 .* G2$

- enter the power to raise $G1_n$ and $G2_n$ to: [1 1] is power of 1
- e.g., [1 1/2] would take square root of $G2_n$ before multiplying with $G1_n$



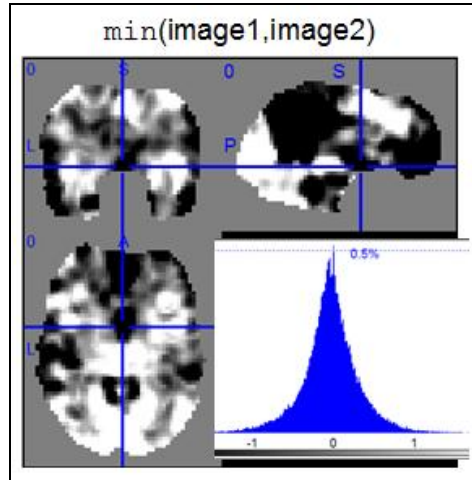
4. divide: $G1 ./ G2$

- enter the power to raise $G1_n$ and $G2_n$ to: [1 1] is power of 1
- e.g., [1 1/2] would take square root of $G2_n$ before dividing $G1_n$ by it



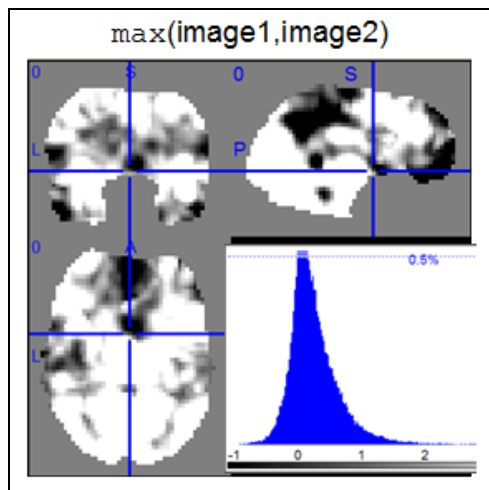
5. *min*: $\min(G1,G2)$

- For the n th image pair ($G1_n$ and $G2_n$), at every corresponding voxel ...
 - the “minimum statistic” takes the lesser value ... e.g., $\min(5,3) = 3$



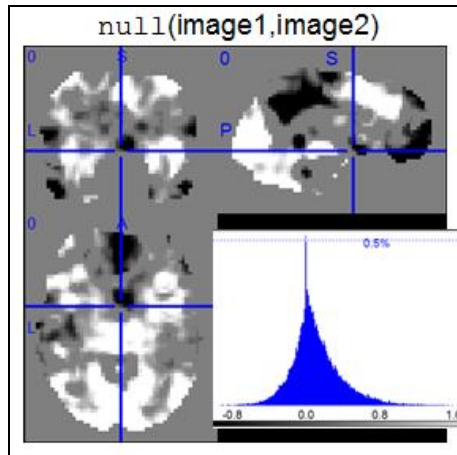
6. *max*: $\max(G1,G2)$

- For the n th image pair ($G1_n$ and $G2_n$), at every corresponding voxel ...
 - the “maximum statistic” takes the greater value ... e.g., $\max(5,3) = 5$



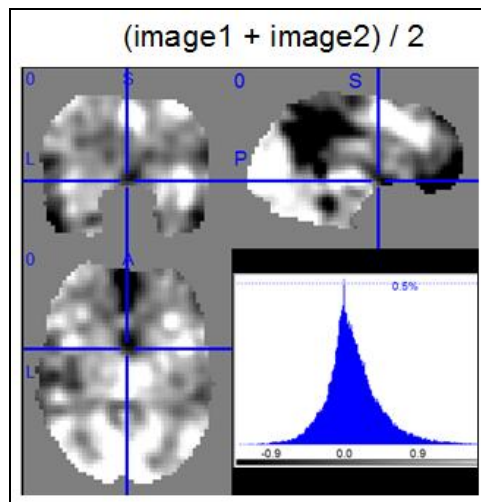
7. *null*: the “null statistic”

- For the n th image pair ($G1_n$ and $G2_n$), at every corresponding voxel i ...
 - If $G1_{n,i} > 0$ and $G2_{n,i} > 0$, null takes the value closer to zero
 - If $G1_{n,i} < 0$ and $G2_{n,i} < 0$, null takes the value closer to zero
 - If the sign of $G1_{n,i}$ and $G2_{n,i}$ don't match, null assumes a value of 0

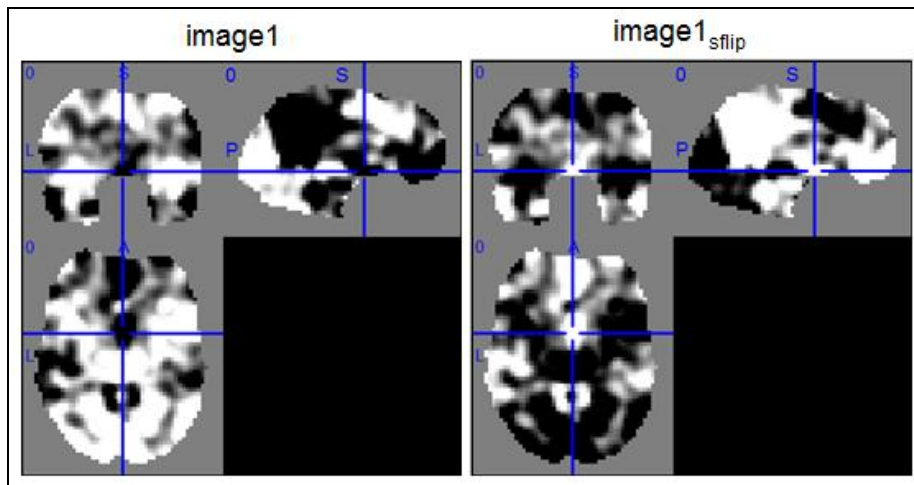


8. *mean*: $(G1 + G2) / 2$

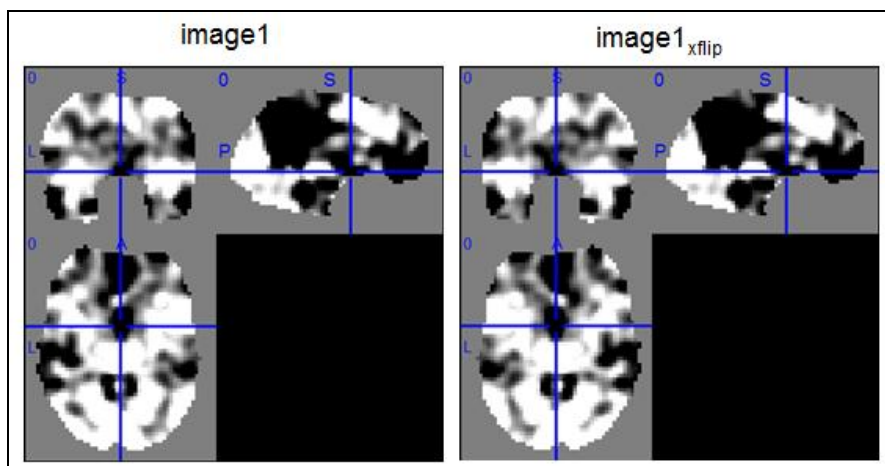
- For the n th image pair ($G1_n$ and $G2_n$), at every corresponding voxel i ...
 - Take the average of the two voxel values



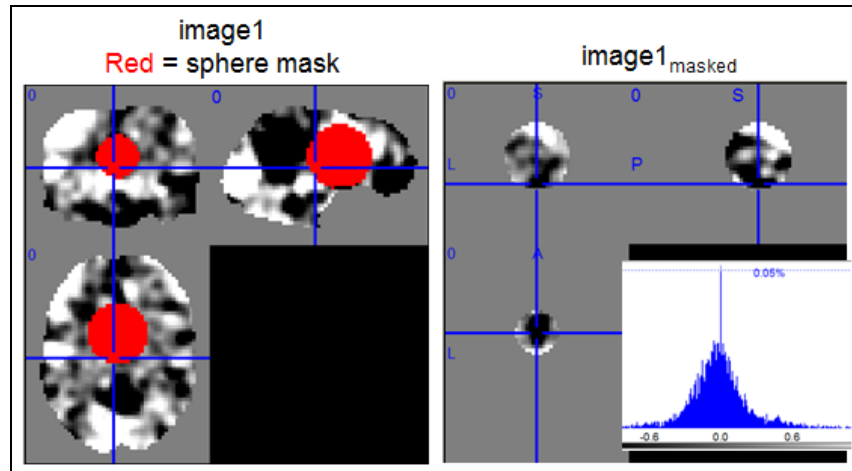
- **[7] flip sign of all non-zero voxels**
 - Select the N images
 - `imcalc` will flip the sign of all non-zero voxels
 - e.g., $+5.2$ becomes -5.2



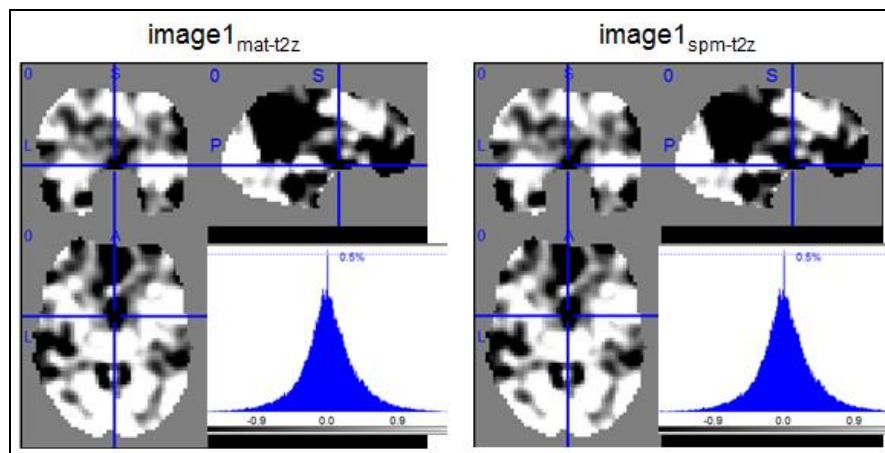
- **[8] flip image along x-axis;**
 - Select the N images
 - For each volume V_n ...
 - `flipdim(Vn,1)` to flip along the x-dimension



- **[9] mask images to a template:**
 - Select the N images
 - Select a single mask image (example: spherical mask)
 - Sample_data \rightarrow mask_sphere.nii
 - imcalc will binarize the mask, and multiply the mask times each image

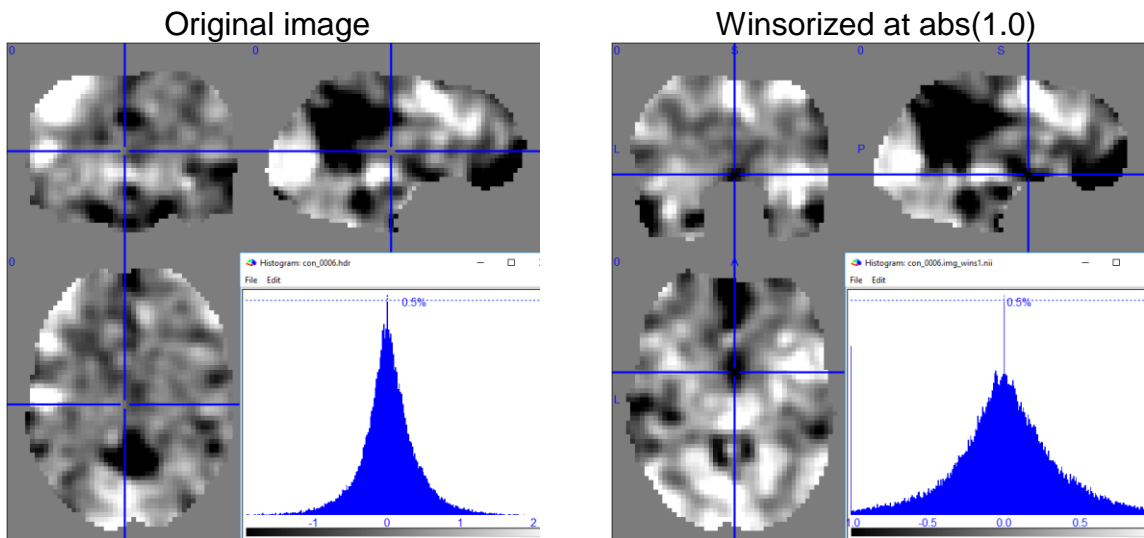


- **[10] T-to-Z transform:**
 - Select the N images
 - imcalc assumes all images have the same degrees of freedom
 - Two options
 1. [1] uses two matlab functions (`norminv` and `tinv`)
 - a. Will only work if the MATLAB “stats” toolbox is installed
 2. [2] uses the spm function `spm_t2z`
 - a. Will work regardless of the “stats” toolbox



- **[11] Winsorize (cap) extreme values**

- Select the N images
- Enter the absolute value of the value at which to cap voxels
- e.g., cap = 5 will turn all voxel values less than -5 to -5 , and all voxel values greater than $+5$ to $+5$
- Example:



(options continue on next page)

Other options, briefly described:

- [12] Z-score transform of image relative to its global mean and SD;
 - write a z-transformed version of the image
- [13] percap: cap at a fixed percentile (upper and lower)
 - similar to [11], but capping values using a percentile threshold
- [14] divide by X times the SD of non-zero voxels
- [15] write single voxels to a .nii
 - user specifies XYZ coordinate (in mm), which is written to an image
- [16] create a cluster image
 - build a cluster ROI based on individual binarized ROIs
- [17] split cluster image(s) into constituent images
 - the reverse of [16]
 - also useful to split apart a brain parcellation image (e.g., [here](#)) containing N uniquely labeled regions into N distinct image files
- [18] linear XYZ translation
 - physically shift non-zero values (e.g., a spherical ROI)
- [19] [deprecated]
- [20] write hemisphere masks from template
 - easy way to get hemisphere masks for a given image size

(options continue on next page)

[21] homotopic voxel calculations

- basic mathematical operations on Left- versus Right-hemisphere voxels at homotopic (i.e., voxels with the same Y- and Z-coordinates, but X-coordinates of opposite sign)

Select the desired operand (O = original; X = x-flipped):

[1] O - X (signed)

[2] X - O (signed)

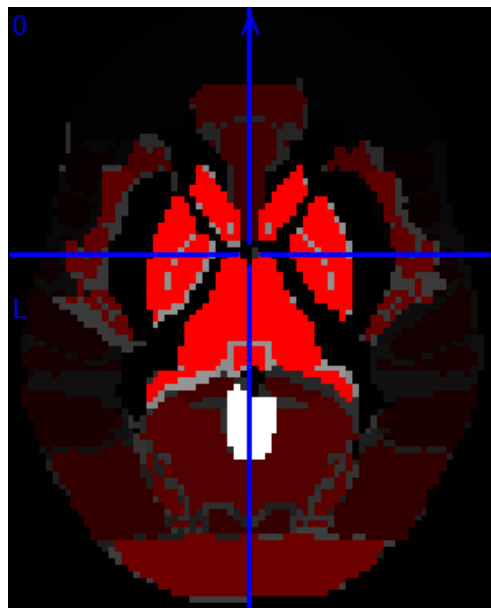
[3] min(O,X)

[4] max(O,X);

[5] mean(O,X);

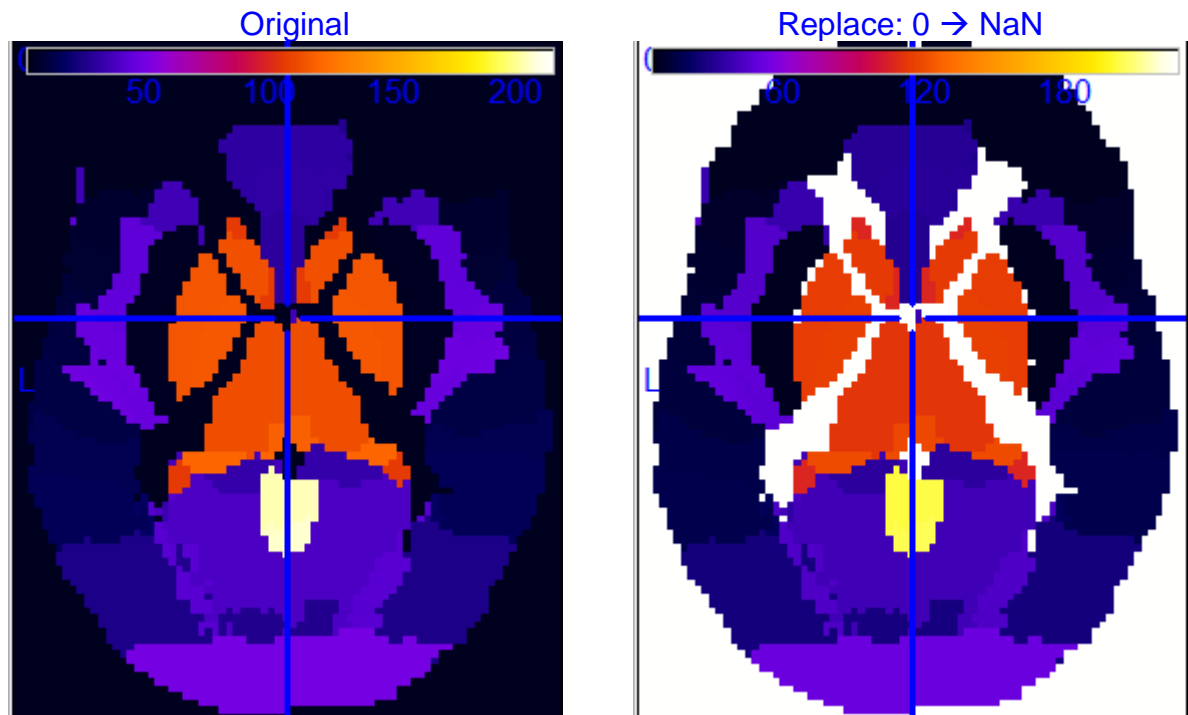
[6] compare(O,X) at homotopic voxels:

- For example, option [6] returns a volume in which 1s indicate identical values in homotopic locations (a useful way to compare the symmetry of anatomical parcellations or other quantized images).
- This is illustrated **below**; the red image shows voxels that have the same parcellation value in L and R voxels; visible gray areas indicate non-overlap between the hemispheres.



[22] replace zeros with ___

- replace zero values with some other value
- an easy way to change image background from black (0-value) to white (high value) for cleaner viewing / slide presentations
- See an example, **below**:



[23] inter-subject proportion maps

- e.g., for the LONI dataset: calculate inter-subject map for a given ROI (e.g., voxels in STG) across available subjects

[24] pad image with extra voxels

- e.g., add 10 empty voxels on each side of the image