

# Frequently Asked Questions

## for [NIFTI and ANALYZE tools](#)

---

### Orientation related question

Is it possible to use `reslice_nii` to apply a transform to a nifti volume? for example, if I have a transformation matrix that is the result of a 3D coregistration of a functional with an anatomical volume, could I use `reslice_nii` to apply that transformation matrix to the functional volume to transform it into anatomical space? If so, any chance of some example code? (30-April-2012, Extracted from Mark from MATLAB File Exchange Comments)

### Answer

Sure Mark. You need a reference image by the way, so you can see what will happen before and after you apply the transformation matrix. Here's the example:

1. Download "avg152T1\_RL\_nifti.nii" from NIFTI site. I use it as reference image.

2. Load and view how the reference image looks like:

```
nii = load_nii('avg152T1_RL_nifti.nii');  
view_nii(nii);
```

3. Assume that I have a transformation matrix, which will let the reference image turn 30 degree counter-clockwise on XY plane, here will be the matrix:

```
T = [cos(pi/6) -sin(pi/6) 0; sin(pi/6) cos(pi/6) 0; 0 0 1];
```

4. Get `old_xyz` from reference image:

```
r1 = load_untouch_nii('avg152T1_RL_nifti.nii');  
old_xyz = [r1.hdr.hist.srow_x(1:3); r1.hdr.hist.srow_y(1:3); r1.hdr.hist.srow_z(1:3)];
```

5. Apply your transformation matrix, and save `new_xyz` into a new image:

```
new_xyz = T * old_xyz;  
r1.hdr.hist.srow_x(1:3) = new_xyz(1,:);  
r1.hdr.hist.srow_y(1:3) = new_xyz(2,:);
```

```
rl.hdr.hist.srow_z(1:3) = new_xyz(3,:);  
save_untouch_nii(rl, 'rl30.nii');
```

Now you already have the "rl30.nii", which is a transformed NIfTI image can be used anywhere.

6. In order to view this image using my toolbox, you need to reslice it:

```
reslice_nii('rl30.nii', 'rl30b.nii');
```

Now, you can load and view the rotated image:

```
rl30b = load_nii('rl30b.nii');  
view_nii(rl30b);
```

---

## Orientation related question

I have some questions regarding load\_nii function when used with nifti files. I am working on a project where the transform matrix saved in the nifti header can be very important. The description of load\_nii says: "A subset of NIFTI transform is included." What is that subset? Does it just applies the (orthogonal) rotational component of the matrix and saves it in img field? (15-March-2012, Extracted from Chitresh Bhushan at University of Southern California, US)

## Answer

The subset of transformation includes flipping and orthogonal rotation with a total of 48 orientations. However, it has nothing to do with "save", since this is for "load" only. When you use "save\_nii" to save your data to NIfTI file, it can be any orientations.

In order to make "load\_nii" easy to use, I actually interpret the header for the users, and make voxel coordinate system (local) align with the millimeter coordinate system (world or global). This way, any user application only need to take care of the orientation of 1 coordinate system, instead of the orientation of 2 coordinate systems.

The reason that "load\_nii" only includes a subset of transformation is that I just want to avoid unnecessary interpolation. The "tolerance" parameter of "load\_nii" serves the same purpose. If you do have data with non-orthogonal rotation, shearing etc., then you can always use "reslice\_nii" to reslice your data before using "load\_nii".

Again, this limitation only applies to "load\_nii". Probably the following example may help you understand this better:

```
nii = load_nii('avg152T1_LR_nifti.nii');
nii.hdr.hist.sform_code = 1;
nii.hdr.hist.srow_x = [-2 1 0 90];
nii.hdr.hist.srow_y = [1 2 0 -126];
nii.hdr.hist.srow_z = [0 0 2 -72];
save_nii(nii, 'tst.nii');
reslice_nii('tst.nii','tst1.nii');
nii1 = load_nii('tst1.nii');
view_nii(nii1);
```

---

## Orientation related question

Saving NIfTI files with `sform = qform = 0` leads to a problem: Many programs (FSL, FreeSurfer), and TrackVis/DTK I believe, maintain backwards compatibility with Analyze 7.5 orientation (LAS) data by assuming "all Analyze files and any NIfTI file which did not specify either `qform` or `sform`, had a reversed left-right orientation" (see <http://nifti.nimh.nih.gov/nifti-1/support/FSLandNifti1>). This assumption is for backwards compatibility, and is not valid when applied to NIfTI datasets written by MATLAB program (in which `sform = qform = 0` means RAS), and hence unexpected L-R flip. (28-January-2011, Extracted from Bryce Wilkins at Biomedical Imaging Lab, University of Southern California, US)

## Answer

First of all, my MATLAB tools never save any file with `sform = qform = 0`. When you see that `sform = qform = 0` in the structure that is loaded by "load\_nii.m", it only means that the affine matrix has been properly interpreted. As a matter of fact, the file written by "save\_nii.m" is always a NIfTI file with `sform = 1` and `qform = 0`, and you can always check the real header information stored in any NIfTI (ANALYZE) file by using "load\_untouch\_nii.m". In other words, you can never save your data to a NIfTI file with `sform = qform = 0`. If you want to save your data to an ANALYZE format, you can do so by using "save\_untouch\_nii.m". In that case, there is neither `sform` nor `qform` in the ANALYZE header.

Second, the link that you mentioned is for FSL only. FSL assumes that all ANALYZE files are in radiological convention (LAS). On the other hand, many other software assume that ANALYZE files are in neurological convention (RAS). Because there is really no orientation information stored in ANALYZE header, the laterality of an ANALYZE file is all based on the assumption. Please see more detail in **Nifti\_tools.pdf**.

---

## Orientation related question

What I was observing is a L/R flip with the normal load\_nii and save\_nii tools. Fslview displays the loaded and saved image in a different orientation ... (11-June-2009, Extracted from Richard Green at Imaging Science and Biomedical Engineering, University of Manchester, UK)

## Answer

As a matter of fact, it is FSL View that flips the ANALYZE image, not my tools. If you are still using ANALYZE image, I strongly suggest that you take a look at the table that I summarized in **UseANALYZE.pdf**.

---

## Orientation related question

I am working with "load\_untouch\_nii". What is the orientation of the XYZ? How can I transfer it into MNI coordinates? (01-June-2009, Extracted from Asaf Stahl at Biomedical Engineering, Tel-Aviv University, Israel)

## Answer

First, I have to let you know that "load\_untouch\_nii.m" is a very special function. Please type: help load\_untouch\_nii, and read the instruction very carefully. You will find that this function only properly reads the header and data; however it does not apply any appropriate affine geometric transform or voxel intensity scaling in the header to the data. Therefore, the user will have the responsibility to do the interpretation by themselves.

If your image file is in ANALYZE format, no one can tell you the XYZ orientation. If your image file is in NIfTI format, please use "load\_nii.m". Because if you use "load\_nii.m", the affine matrix is always transformed to a diagonal matrix with positive entries and the image matrix in the loaded structure will be re-oriented correspondingly, which is actually in neurological convention or RAS.

If there is AC Origin information in the header, you can easily obtain the coordinate in millimeter by subtracting the origin location and multiplying the voxel size; otherwise, you won't be able to obtain "coordinate in millimeter" correctly. In addition, both "MNI coordinate" and "Talairach coordinate" are brain atlas. So I believe that you are mean "coordinate in millimeter" by saying "MNI coordinate".

---

## Orientation related question

I want to know how I could make a NIfTI file which is in LAS orientation. The reason why I need to use LAS is because my output files use that; also the functions I use are batch processes, so each image that is created has to be switched from RAS to LAS manually. (08-September-2008, Extracted from Felipe S. Salinas at Biomedical Engineering, UT Health Science Center San Antonio, US)

## Answer

Program "flip\_lr.m" is provided to do LAS->RAS (or RAS->LAS) conversion for any NIfTI or ANALYZE file format. However, please make sure that you save those flipped NIfTI file separately, since you may confuse yourself if you mix them with other normal NIfTI file.

---

## Orientation related question

When we use "load\_nii.m" to process our data containing non-orthogonal affine transform matrix, the values are not correct since we have compared them with MRICroN. So, how to deal with the data with such an affine matrix? (21-June-2008, Extracted from Chaogan Yan at Laboratory of Cognitive Neuroscience and Learning, Beijing Normal University, China)

## Answer

When you use "load\_nii.m" to load your NIfTI data, we usually allow 10% of distortion tolerance by default. If it is too oblique, it will notify you. You can adjust the tolerance parameter when you use "load\_nii.m". For example, you can set it to 0, so that your NIfTI image will not be loaded if it contains any slightly non-orthogonal rotation.

For those NIfTI file that cannot be loaded with "load\_nii.m", you should use my "reslice\_nii.m" to reslice your NIfTI image, which is exactly what MRICroN does. It will not cause negative effect, as long as you remember not to do *slice time correction* after using "reslice\_nii.m".

---

## Orientation related question

According the NIfTI standard, xyz's world space coordinates are always in RAS, but ijk's voxel space coordinates can be in any orientation. So what your program actually seems to be doing is to take the transform matrix from the header and re-index each voxel so that voxel indices in ijk's space conform to RAS. Is it right? (28-June-2007, Extracted from Samuel A. Hurley at Applied Neuro MRI Lab, University of Wisconsin, US)

## Answer

Yes. This is exactly what the transformation does in my programs. It actually transforms the affine matrix in the header to a diagonal matrix with positive entries and re-orient the image matrix in the loaded structure correspondingly. i.e. to let the image matrix field of the loaded structure be in neurological convention or RAS.

If the transformation only includes several flips and orthogonal rotations, the "load\_nii.m" can do it; otherwise, you can always use "reslice\_nii.m" to do the transformation with non-orthogonal rotation or shearing.

After the transformation is done, ijk will be aligned to xyz respectively. Therefore, the relationship between ijk & xyz is now simplified to just a scaling factor of "hdr.dime.pixdim(2:4)", and it is much easier to tell the location a voxel that is close to left, right, anterior, posterior, superior, or inferior.

---

## Orientation related question

I noticed a problem when I use save\_nii to write out a NIfTI file. Basically, I tried reading a NIfTI file and then write it out without making any changes, I found the "flip\_orient" and "rot\_orient" in the "hdr.hist" of the NIfTI structure become empty in the new .nii file that is written out by save\_nii. Could you please let me know what the problem is and how to fix it if possible? (14-December-2006, Extracted from Ying Guo at Rollins School of Public Health, Emory University, US)

## Answer

This is not a problem.

First, "flip\_orient" and "rot\_orient" fields in the "hdr.hist" are used internally for information only, which indicate how the transformation is done. If you find that there are "flip\_orient" and "rot\_orient" fields in the "hdr.hist" of the loaded structure and they were not empty, it means that the file you are loading is a NIfTI format file, and the affine matrix is not a diagonal matrix with positive entries. In addition, fields like "hdr.hist.new\_affine", "hdr.hist.old\_affine", and "original" are also for information only.

Second, the new NIfTI file that you saved is changed by the program, because the NIfTI header information is appropriately applied to the data. The new NIfTI header is also changed to indicate that its information has been properly applied. e.g., the affine matrix is now a diagonal matrix with positive entries. When you use "load\_nii.m" again to load the new NIfTI file, there is nothing to flip or rotate. So the "flip\_orient" and "rot\_orient" fields in "hdr.hist" of the NIfTI structure is empty.

If you would like to save exactly as loaded, you can use "load\_untouch\_nii.m" to load a NIfTI or ANALYZE file. In this case, nothing is changed. That means, you have to interpret the header information and apply them to the image matrix by yourself. Therefore, you cannot view or save the untouched data with "view\_nii.m" or "save\_nii.m" program. You can only use "save\_untouch\_nii.m" to save this loaded structure.

---

## Orientation related question

When I load my data using your NIfTI tools, which side is left and which side is right? Was my data stored in RAS orientation? (23-February-2006, Extracted from Katja Osswald at Brain & Body Centre, University of Nottingham, UK)

## Answer

The answer is that it depends on your data.

If your data is in old ANALYZE 7.5 format, this tool cannot tell if an ANALYZE data is in RAS or LAS orientation. You may notice that strict ANALYZE 7.5 format has a field called "hist.orient". The value of this field ranges from 0 to 5 covering 6 orientations. However, many other important orientations (e.g. images in neurological convention or RAS) can not be expressed using Strict ANALYZE 7.5 format, which

poses a critical problem for this "hist.orient" field. You can find more details from: <http://www.grahamwideman.com/gw/brain/analyze/formatdoc.htm>. Therefore, many software (e.g. PLSGUI, SPM, AFNI, FSL, MRICro, etc.) choose to ignore this field, and the orientation will totally depend on the ANALYZE image data itself. If you are a radiologist, you can assume your data are in LAS orientation; if you are a neurologist, you can assume your data are in RAS orientation. In order to be compatible with ANALYZE images saved by other software, this tool will also ignore the "hist.orient" field.

However, since the NIfTI image includes affine matrix, this tools can tell how the data is stored. For example, if the affine matrix is a diagonal matrix with positive entries, the data is stored in RAS orientation.

There are two example datasets, "avg152T1\_LR\_nifti.nii" and "avg152T1\_RL\_nifti.nii", are provided under NIfTI web site <http://nifti.nimh.nih.gov/nifti-1/data>. As is indicated on NIfTI web site, "*The first image (LR) is stored in radiological convention (LAS). The second image (RL) is stored in neurological convention (RAS). Any NIfTI compliant viewing software should display these images identically*".

Therefore, there could be 3 situations when you use the command: `nii = load_nii('filename.nii');`

1. If you find "nii.filetype" is 0, it means that the orientation is whatever in your image data, because your data is in old ANALYZE 7.5 format. I can neither tell you which side is left (right), nor tell you your data was stored in which orientation.
2. If "nii.filetype" is not 0, and there are no "rot\_orient" / "flip\_orient" fields under "nii.hdr.hist" or if they are empty, it means that the affine matrix was a diagonal matrix with positive entries, and your data was stored in RAS orientation. The image matrix field of the loaded structure is already in RAS orientation (left side is left, right side is right).
3. If "nii.filetype" is not 0, and "rot\_orient" / "flip\_orient" fields under "nii.hdr.hist" are not empty, it means that the affine matrix was not a diagonal matrix with positive entries, and it is transformed to such one and the image matrix is re-oriented correspondingly. So the image matrix field of the loaded structure is now in neurological convention or RAS orientation (left side is left, right side is right).

---

## Image overlay related question

I created an image using `make_nii`, and saved this image as a .nii file, and tried to overlay it on a template image in MRICron, but the resulting .nii image was very very small. Do you happen to know if there is any way to correct this so that I can overlay it properly? (20-November-2013, Extracted from Victoria McLelland at Memory and Perception Lab, University of Toronto, Canada)

## Answer

This is because the voxel size of your image is larger than that in your template image. The following workaround should work:

1. Get dimension and voxel\_size from the template image.
2. Run command: `reslice_nii('current.nii', 'new.nii', voxel_size);`
3. Since the dimension of image will change after you reslice it, you need to run `pad_nii` and/or `clip_nii` afterwards.
4. Here is an example:

```
b = load_nii('new.nii');
t = load_nii('ch2bet.nii'); % template
b_org = b.hdr.hist.originator(1:3)
b_dim = b.hdr.dime.dim(2:4)
t_org = t.hdr.hist.originator(1:3)
t_dim = t.hdr.dime.dim(2:4)
% Since your b_dim is smaller than t_dim, we only need to run pad_nii
% otherwise, you need to run clip_nii (after pad_nii, or alone)
option.pad_from_L = t_org(1)-b_org(1);
option.pad_from_R = (t_dim(1)-t_org(1))-(b_dim(1)-b_org(1));
option.pad_from_P = t_org(2)-b_org(2) ;
option.pad_from_A = (t_dim(2)-t_org(2))-(b_dim(2)-b_org(2));
option.pad_from_I = t_org(3)-b_org(3);
option.pad_from_S = (t_dim(3)-t_org(3))-(b_dim(3)-b_org(3));
b2 = pad_nii(b,opt);
save_nii(b2, 'new2.nii');
% if you overlay new2.nii to template, now they should match
```

---

## Image overlay related question

How can I get a plot that look like the screenshot in NIfTI web page? (15-February-2011, Extracted from Jakir Hossen at University of Memphis, US)

## Answer

Here is how I plot an EEG source imaging on top of T1 background:

1. Download overlay.zip from [LINK HERE](#).
2. Type the following MATLAB command:

```
T1 = load_nii('T1.nii');  
EEG = load_nii('EEG.nii');  
option.setvalue.idx = find(EEG.img);  
option.setvalue.val = EEG.img(option.setvalue.idx);  
option.useinterp = 1;  
option.setviewpoint = [62 48 46];  
view_nii(T1, option);
```

---

## Image overlay related question

I have been using your program "Tools for NIfTI and ANALYZE image". I try to map a "color.nii" on "background.nii". With FAQ on the website, I still got errors and I do not know how come. Can you help me please? (04-April-2007, Extracted from Karla De Smedt at Katholieke University Leuven, Belgium)

## Answer

The cause of the problem is from your "color.nii" data. when you use "opt.setvalue.idx = find(map.img>0)" to find index, the result is empty, since there is one -1 and all 0 in that file. In addition, you will need at least 2 different colors in "color.nii" in order to display properly. Hopefully this will clear your problem.

---

## Image overlay related question

I have been trying to overlay my activation map with a jet colormap onto a background MRI (with a gray colormap) just as in the opening figure in the webpage, and I can't seem to find a way to do it. On the FAQ it only explains how to overlay more than two images. (13-September-2006, Extracted from Jos é Rebola from jrebola@gmail.com)

## Answer

There are 2 FAQ regarding to overlay, and the other one is much closer to yours. Let me customize that one into your case, and here is how to use this advanced feature:

1. Use command below to load your background image to "bg" struct:

```
bg = load_nii('background_image.img');
```

2. You need to know the threshold of your activation map, and the index of the thresholded activation map. Assuming that your threshold is 0, you can use:

```
map = load_nii('activation_map.img');
```

to load your activation map to "map" struct, and extract its value and index to "opt" struct:

```
opt.setvalue.idx = find(map.img);  
opt.setvalue.val = map.img(find(map.img));
```

You can see more detail to use "setvalue" by command:

```
help view_nii;
```

3. Add other values to "opt" struct:

```
opt.command = 'init';
```

4. If you want to have an interpolated view, also set:

```
opt.useinterp = 1;
```

5. Plot activation map with background image by command:

```
view_nii(bg,opt);
```

Please don't click "interp on/off" menu in the view window, because it will not work properly under this situation. Instead, you should use:

```
opt.useinterp=0;
```

or:

```
opt.useinterp=1;
```

6. In the view window, you can change the colormap of your activation map by selecting "Jet".
- 

## Image overlay related question

Could you please tell me: (1) How to overlay 2 or 3 other images together? (2) How to choose different threshold? (14-May-2006, Extracted from Ruiwang Huang at Forschungszentrum Jülich, Germany)

## Answer

You can always overlay 2 or more other images together manually with the following simple procedures, if dimensions of all the images are the same. Please do not click "Interp on" when you view the integrated image, because it will be a mess if the interpolation could not be separated.

In addition, if the datatypes are different, you must use "make\_nii" to convert a datatype to the highest storage. For example, after you load a background image with `bg = load_nii('background_image.img')` and find the `bg.img` is `uint8` (i.e. `datatype = 2`), but some image is `single` (`datatype = 16`), you need to convert `bg` to `single` datatype with `bg = make_nii(single(bg.img), bg.hdr.dime.pixdim(2:4), bg.hdr.hist.originator(1:3), 16)`.

How to choose different thresholds depends on which regions of your activation image you would like to display. For example, if the values in your first activation map range from `[-6 +7]`, and you want to view regions whose values are from `[-6 -2]` and from `[+3 +7]`, you can use the following procedures:

1. Load your first activation map:

```
map1 = load_nii('activation_map1.img');
```

2. Get index of that activation map:

```
idx1 = find(map1.img < -2 | map1.img > +3);
```

3. Get `map2 / idx2` etc. in the same way.

4. Load background image:

```
bg = load_nii('background_image.img');
```

5. Make sure bg.img and all map.img must have the same dimension & datatype.

6. Overlay the 1st layer above the background:

```
bg.img(idx1) = map1.img(idx1);
```

7. Overlay the layer above till the top layer:

```
bg.img(idx2) = map2.img(idx2);
```

8. Display the integrated image:

```
view_nii(bg);
```

---

## Image overlay related question

I have been using your NIfTI toolbox for MATLAB recently and was wondering whether it is possible to make an overlay of a thresholded image onto an ANALYZE image. In effect that would mean that all 0 values of the thresholded image should be transparent so that the underlying functional image can be seen. (13-February-2006, Extracted from Henry Lütcke at Biomedical NMR Research, Max Planck Institute for Biophysical Chemistry, Germany)

## Answer

This is an advanced feature, and it is already included in my tools.

Here is the brief guide to use this feature:

1. Use command below to load your ANALYZE image to "ana" struct:

```
ana = load_nii(ana.img);
```

2. Load your thresholded image to "thresh" struct:

```
thresh = load_nii(thresh.img);
```

and extract its value and index to "opt" struct:

```
opt.setvalue.idx = find(thresh.img);
```

```
opt.setvalue.val = thresh.img(find(thresh.img));
```

You can see more detail to use "setvalue" by command:

```
help view_nii;
```

3. Add other values to "opt" struct:

```
opt.command = 'init';  
opt.useinterp = 0;
```

4. Plot thresholded image with overlaid ANALYZE image by command:

```
view_nii(ana, opt);
```

5. Don't click "interp on/off" menu. It doesn't work properly. If you would like to see the interpolated one, use:

```
opt.useinterp = 1;  
view_nii(ana, opt);
```

---

## Image overlay related question

I am trying to overlay a ROI file onto a template file. I have 112 cortical and subcortical areas for parcellation, using the Harvard-Oxford atlas and project them on the template when they show significant differences on a certain parameter (e.g. clustering coefficient) between my two groups, so I can directly view the areas. I only see the Template, but not the ROI. Can you help me out? (17-July-2013, Extracted from Zsuzsi Sjoerds at Department of Psychiatry, VU University Medical Center, Netherlands)

## Answer

1. The voxel size and dimension in template image must be the same as those in ROI image. Right now in template image the voxel size is [0.5 0.5 0.5] and the dimension is [301 370 316], while in ROI image the voxel size is [2 2 2] and the dimension is [91 109 91]. Script below is an example to show you how the voxel size and dimension in template image is changed to the same size as those in ROI image:

```
% change voxel size to make it the same as those in ROI  
reslice_nii('ch2better.nii.gz','bg.nii',[2 2 2])
```

```
% check new template  
bg = load_nii('bg.nii') % it tells me new dim is [76 94 80]
```

```

bg.hdr.hist.originator          % it tells me new origin is [38.5 54.5 36]
% let's make it [39 55 36]

% check ROI again
ROI = load_nii('ROIsHarvardOxford-cort-maxprob-thr25-2mm_YCG_100.nii')
ROI.hdr.hist.originator        % it tells me ROI origin is [46 64 37]

% since new template is smaller, we need to pad it:
img = bg.img;
img = cat(1,zeros(7,94,80),img); % pad X1, 7 is from 46-39
% now we get new img [83 94 80]
img = cat(2,zeros(83,9,80),img); % pad Y1, 9 is from 64-55
% now we get new img [83x103x80]
img = cat(3,zeros(83,103,1),img); % pad Z1, 1 is from 37-36
% now we get new img [83x103x81]

% since our target dimension is [91 109 91] we continue pad it:
img = cat(1,img,zeros(8,103,81));
% now we get new img [91x103x81]
img = cat(2,img,zeros(91,6,81));
% now we get new img [91x109x81]
img = cat(3,img,zeros(91,109,10));
% now we get new img [91x109x91]

% now we can make new template
Template = make_nii(img);

```

2. The overlaid image should have more than 2 colors. Since yours is a binary ROI image, here's what you need to add to the beginning of your script:

```

ROI.img = single(ROI.img);
ROI.img(1) = 0.1;

```

3. Now, the rest of your script should work without any problem:

```

option.setvalue.idx = find(ROI.img);
option.setvalue.val = ROI.img(option.setvalue.idx);
option.useinterp = 1;

```

```
option.setViewpoint = [15 49 38];  
view_nii(Template, option);
```

---

## RGB color format related question

I want to change the pixel value of a xyz vector image by taking its absolute values. I first used: `nii = load_nii(1_xyz.img)`, if I use `view_nii(nii)`, I found everything correctly. Moving the crosshair to 139,63,91, I can read the value is (-0.1784,0.8963,0.3663), which is correct. But if I used: `a = nii.img; a(139,63,91,1:3)`, it will display values as 0.35555,0.9434,0.6535. It seems that the value of a is always positive. I tried to use `nii = load_untouch_nii(1_xyz.img); a = nii.img`; It has the same positive value. Do you have any suggestion about this inconsistency between the value of `nii.img` and that displayed by `view_nii(nii)`? (10-July-2012, Extracted from Xiaoying Tang at Center for Imaging Science, Johns Hopkins University, US)

## Answer

I received the image, and it is a special RGB data type (i.e. `datatype=511`, `bitpix=96`, `float32 RGB`), which is defined by software MRImstudio (<https://www.mristudio.org>).

It requires a special way to display the voxel value. The `nii.img` ranges from 0 to 1 as usual, but the output value needs to be scaled by `nii.hdr.dime.glmax & glmin` in this way:

```
R = img(i, j, k, 1) * (glmax-glmin) + glmin;  
G = img(i, j, k, 2) * (glmax-glmin) + glmin;  
B = img(i, j, k, 3) * (glmax-glmin) + glmin;  
V = [R G B]
```

---

## RGB color format related question

In MATLAB, I have created an overlaid image with RGB values. The matrix `mydata` has four dimensions, corresponding to [x y z RGB] (The size of the fourth dimension is 3). When I am trying to save this data into an RGB NIfTI file, and using your command:

save\_nii(make\_nii(mydata,[],[],128),'mydata.nii'), the resulting NIfTI file cannot be opened with the FSLView program (I get a segmentation fault error). Probably I am doing something simple wrong? (9-June-2011, Extracted from Audrey Peiwen Fan at Martinos Center for Biomedical Imaging, Harvard University, US)

## Answer

I just implemented this feature. However, you will still not be able to view your resulting NIfTI file with FSLView, if it is version 2.4.0b or less. I tested RGB NIfTI data with FSLView 2.4.0b, and it simply does not support this datatype, although there is no segmentation fault error. If you are using SPM8 or less, it does not support RGB NIfTI data either. MRICron only supports old RGB NIfTI data, if you are using Version 4 (April 2011) or less. I tested RGB NIfTI data with this version, and it sorts slice in old RGB way. i.e. if you use command: save\_nii(make\_nii(mydata,[],[],128),'mydata2.nii',1), 'mydata2.nii' can be viewed by MRICron properly, but it is not complied with NIfTI format. Till now, AFNI is the only program that I found complies with NIfTI format. When you view 'mydata.nii' with AFNI, it will display properly. If you view 'mydata2.nii' with AFNI, you will find that the slice is not arranged properly, which is correct because 'mydata2.nii' itself is wrong.

If load\_nii, make\_nii, save\_nii, and view\_nii are the only commands that you need, you can simply use my NIfTI tool to handle RGB NIfTI data.

---

## Program “reslice\_nii” related question

I tried reslice\_nii option as recommended, but it changes the dimension of image file. For example, my original data was 64\*64\*31 but resliced image has 74\*80\*59.. Why does it change the dimension of file? Isn't it possible to not change the dimension? (02-April-2012, Extracted from Misun Kim at Brain Reverse Engineering and Imaging Lab, KAIST, Korea)

## Answer

This is obvious. Think about a unit square in 2D plane with dimension of [1 1]. If you rotate it 45 degree, you have to use a dimension of [1.414 1.414] to hold the same square. However, the distance from origin to edges are still 1.

---

## Program “reslice\_nii” related question

When I working with reslice\_nii.m function, the problem that I'm facing is that when I input an image (e.g. 256,256,26 with pixdim 0.97 0.97 4.9), giving the new pixdim (0.97 0.97 2.16), the output image dimensions are modified (e.g. 259,263,53). Do you know what could I do? (27-April-2011, Extracted from Jose Angel Pineda Pardo at Center for Biomedical Technology, Technical University of Madrid, Spain)

## Answer

Think about a unit cubic, and imaging it only rotate 45 degree along the z axis. The dimension will change from [1 1 1] to [1.414 1.414 1]. You know, the affine matrix can be much more complicated, and you can never foresee the new dimension.

However, you can specify any new pixdim. By default, it will give you isotropic data, but you can specify any, let's say, you can still keep your old pixdim of [0.97 0.97 4.9].

---

## Image editing related question

I would like to modify the value of the masks I have created for my ROIs. I have downloaded your package, and successfully load et view one of my NIfTI file, but I don't know how to change the value of the mask (e.g. from 1 to 2). (22-June-2011, Extracted from Adeline Dubois at Laboratoire d'Imagerie Fonctionnelle, France)

## Answer

There are two ways to edit voxel value.

1. Using "view\_nii.m" program:

You can use "view\_nii.m" as an individual program. Navigate to the voxel that you want to change its value, and the crosshair will be moved to that voxel. Click "Edit" menu and then choose "Edit voxel value at crosshair", the current voxel value will be displayed. Type the new number within the data type range. The data type can be obtained by click "Image Information" under "View" menu. For example, for 8-bit unsigned integer, its data type range is [0 255]. Now, you can see that the voxel gray level is changed. Finally, click "Save displayed image as" under "File" menu, you can have the modified image saved in NIfTI format.

## 2. Using command line:

Sometimes, you may want to change a lot of voxels to the same value. In this case, it's better to do it in command line. First, load the original image: `nii = load_nii('old_image.nii');` Then: `nii.img(find(nii.img == old_value)) = new_value` will change all `old_value` to `new_value`. Finally, `save_nii(nii, 'new_image.nii')` will save the modified image into 'new\_image.nii'.

For your particular case, I suggest you use command line to do so.

---

## Header editing related question

I have a programme which requires that I update the description section of the .hdr file. Using `load_untouch_header_only.m` I am able to load the header and edit the line required, however I am having some difficulty in trying to save the edited header file. I have tried using `save_untouch_nii_hdr.m`, but I am a little unsure what I should be using as the fid input of the function, since I don't generate one using the `load script..` (8-August-2013, Extracted from Joanne Hale at The University of Birmingham, UK)

## Answer

Program "`load_untouch_header_only`" let you take a look at the header structure without loading the data. If you want to edit the header, you should use "`load_untouch_nii / save_untouch_nii`" pair. Here's an example:

```
nii = load_untouch_nii('avg152T1_LR_nifti.nii');
nii.hdr.hist.descrip = 'Joanne';
save_untouch_nii(nii, 'new_avg152T1_LR_nifti.nii');
```

Although the data is loaded and saved, it will not be touched, and only the header is changed. If the input file is in Analyze 7.5 format, the output is still in Analyze 7.5 format. If the input is in NIfTI format, the output is still in NIfTI format.

---

## Format conversion related question

The data that I need to process is in NIfTI format. If I just use FSL's flirt immediately, the orientation gets reversed, effectively "flipping" the image. I use load\_nii to read in the NIfTI file, and then set filetype in the structure to 0. But when I then use save\_nii and run fslhd on the output image, it is still reading it as a NIfTI image. Is there a way to treat the output as true ANALYZE 7.5 data and not NIfTI img/hdr pairs? (08-October-2010, Extracted from Shane Steinert-Threlkeld at Johns Hopkins University, US)

## Answer

My tool is not designed to convert NIfTI to Analyze7.5, although it can properly read both format.

Especially, 'save\_nii.m' only save data into NIfTI format (with either .nii or .img/.hdr extension), even it was the Analyze data that is loaded by 'load\_nii.m'; while 'save\_untouch\_nii.m' only save data into its original format that is loaded by 'load\_untouch\_nii.m'. Some people may think all files with .img/.hdr extension are Analyze data, but it is not correct.

For your situation, since you know that you will get L/R flipped if you use FSL's flirt directly on your NIfTI image, you can simply use my 'flip\_lr.m' tool to save another flipped copy of your image, and then apply FSL's flirt on the flipped copy.

By the way, it is invalid to set filetype to 0, since it can only be 1 or 2 when you use 'save\_nii.m'. In addition, it is only determined by the filename that you entered (with .nii or with .img/.hdr) rather than by setting filetype field manually.

---

## SPM .mat related question

I'm a student here at UCLA and I've recently started trying to learn how to apply motion correction to fMRI images. Anyway, I learned how to do so using SPM8, but I can only do it with the .nii files that I have. I also have some 4-D Double images of a rat's brain that I would like to apply motion correction to. However, these images are .mat format and SPM8 will not recognize them. Do you know how I can convert from .mat to .nii? I used mat\_into\_hdr.m, which I thought would create a .hdr file, but it only created a variable on MATLAB. I then tried using save\_nii.m, but I just got errors saying it could not open file. I might be formatting it wrong. What should I do? (12-November-2009, Extracted from Rodolfo Correa at University of California Los Angeles, US)

## Answer

If "... images are .mat format ..." like you said, it is probably not the .mat that "mat\_into\_hdr.m" expected. The .mat file that "mat\_into\_hdr.m" expected should be:

1. It should have the same file name as its .img/.hdr counterpart. i.e. something like file1.img / file1.hdr / file1.mat.
2. When you load such .mat file, there is nothing but one variable, called M, which is a 4x4 matrix. It is an affine matrix.

For the .mat that you have, since you said that it is a 4-D image, it should have a 4-D matrix inside. When you load this .mat file, you should see this 4-D matrix. In order to convert it to .nii file, you also need the header information. Once you have both 4D matrix and the header information, you can add them to a variable, like: nii.img = 4-D matrix; nii.hdr = Header Info. Now you can use my "save\_nii.m" program to create a .nii file.

---

## SPM .mat related question

We have noticed that save\_untouch\_nii, when given an Anz data structure (obtained from doing load\_untouch\_nii on an appropriate Anz volume), also saves out a .mat file, that just seems to be a diagonal matrix of the voxel sizes, eg a voxel to mm transform. Why is that being saved? Is there an option to turn that off in case it confuses someone? (25-March-2009, Extracted from Kate Fissell at Learning Research and Development Center, University of Pittsburgh, US)

## Answer

The saved .mat file was invented by SPM software (<http://www.fil.ion.ucl.ac.uk/spm/software>) and is also used by other software that are derived from SPM. It is a smart remedy for the poor design of ANALYZE 7.5. So, like NIFTI, you can easily handle the Left / Right issue, and make correction for the oblique scan. The filename of the .mat file must be the same as the .img/.hdr file (except extension). Inside .mat file, there is only one variable, and it must be called M, which is a transformation matrix. Since the .mat file is very small, and can easily be identified and removed, a .mat file will always be saved together with .img/.hdr scan. If you use load\_nii to load a .img/.hdr scan, the program will always check if there is a .mat file with the same filename in the same folder.

When you use ANALYZE format, the .mat file is necessary for load\_nii, save\_nii, reslice\_nii etc. programs, the transformation in .mat file will be properly interpreted. However, for load\_untouch\_nii, save\_untouch\_nii etc. programs, since they are actually not handling this transformation matrix, it is not necessary. Thank you for the feedback, the .mat file will no longer be saved by save\_untouch\_nii.m.

---

## Originator related question

My question is regarding the AC origin for make\_nii. First of all, I presume the coordinates should be in mm, like the voxel size. Then, I was wondering if AC origin refers to the position of the central slice. If so, for the 3rd coordinate (referring to axial slices), does negative (e.g. -20) indicate an upward translation (i.e. more superior slice)? This is the convention that I'm used to, but I am not sure it applies to make\_nii as well. (4-March-2009, Extracted from Nicole Pelot at Dalhousie University, Canada)

## Answer

Origin parameter in "make\_nii" is in voxel, which is relative to the first voxel of 'img' parameter, i.e. `img(1,1,1)`;

Yes, we know AC Origin means [0 0 0] in millimeter. However, you must tell the NIfTI data where is your [0 0 0] millimeter, so other people can also know it when they open your NIfTI data. Therefore, parameter 'origin' can only be expressed in voxel to locate the exact position of your 3D matrix to be the AC Origin.

---

## Originator related question

I am trying to set the origin of my images to the centre, so that if I read them into SPM2 there is some overlap between my images and their templates. At the bottom of this email I have copied my header fields in full. When I load my image the first time, I noticed that "one\_tp\_nii.hdr.hist.originator" is [32 32 15 0 0], and "one\_tp\_nii.original.hdr.hist.originator" is also [32 32 15 0 0]. After I write the image out with "save\_nii(one\_tp\_nii, 'temp')" and then read it in again with "image\_nii = load\_nii('temp')", "image\_nii.hdr.hist.originator" is still [32 32 15 0 0], but "image\_nii.original.hdr.hist.originator" array is now [256 0 0 0 0]. SPM2 takes [256 0 0] as the origin. Has "save\_nii.m" not written out the originator, or is there another problem with my header? (13-April-2007, Extracted from Simon Robinson at Functional NeuroImaging Laboratory, University of Trento, Italy)

## Answer

First, when you loaded your image, and found that both "hdr.hist.originator" and "original.hdr.hist.originator" are [32 32 15 0 0], it means that the image you loaded is an ANALYZE format image. After you wrote the loaded image with "save\_nii.m", the new image is now in NIfTI format. That is why you see "image\_nii.original.hdr.hist.originator" array is [256 0 0 0 0], because unlike ANALYZE image, NIfTI image does not explicitly have a field called "originator". Please note that the "image\_nii.original" field is what explicitly stored in the image header (except the originator), which is for your reference only, and you should really don't have to worry about it. "image\_nii.hdr" is the one that you have to look at. In your case, "image\_nii.hdr.hist.originator" is still [32 32 15 0 0].

Second, the reason that SPM2 takes [256 0 0] as the origin is because all earlier versions of SPM (before SPM5) can not handle NIfTI image. If the NIfTI file extension is ".nii", it can not be recognized. If the NIfTI file extension is ".img / .hdr", the affine matrix where the originator is defined will not be interpreted.

Now, when you want to save your image into ".img / .hdr" file extension, the "save\_nii.m" in my tools will also save an extra affine matrix into a ".mat" file, so earlier versions of SPM can also open it with correct originator and orientation. If you have an SPM generated image file with its corresponding ".mat" affine matrix file, the "load\_nii" in my tools will also be able to load that affine matrix and integrate it into the .hdr structure.

---

## **NIfTI data dimension related question**

I could not find anything in your code which allows one to incrementally read/write single slices within multi-volume files. e.g. `spm_read_slice()` or `spm_write_plane()`. This is clearly a good idea if one is restricted by RAM and doesn't want to read/write humungous 4D volumes at once. In addition, I could not find a way to save a single volume out of many. (29-January-2009, Extracted from Ashish Raj at Radiology Image Processing Laboratory, Weill Medical College of Cornell University, US)

## **Answer**

For a 4D volume, you can use "expand\_nii\_scan" to save the volume range that you specified to each individual 3D volumes, and use "collapse\_nii\_scan" to save multiple 3D volumes back to a single 4D volume after processing certain 3D volumes.

Generally speaking, you can always use "load\_untouch\_nii" (like "spm\_read\_slice") to read any slices within an any dimension file, and use "save\_untouch\_slice" (like "spm\_write\_plane") to write the loaded and processed matrix back into the original file.

---

## **NIfTI data dimension related question**

I would write feedback for the limitations that I have found, and hope they will be included in a new version. It would be fantastic to access the 5th, 6th and 7th dimensions so I can neatly store my multi-channel multi-echo phase and magnitude MR as (x, y, z, echo, channel, phase/magnitude) in a single file. In addition, would it be possible to read a particular slice z of a 4D NIfTI data (:,:,z,:) and still carry out the image transformation? (19-October-2007, Extracted from Simon Robinson at Functional NeuroImaging Laboratory, University of Trento, Italy)

## **Answer**

I have extended Load / Save / Make programs to N-Dimension, where N can be from 3 to 7 for NIfTI data and from 3 to 4 for ANALYZE data. The size of data, however, will be a big concern. Each element on the 5th dimension will take the size of a 4D time-series image, not to mention the 6th or 7th dimension. This is especially a concern when your data is in 32-bit or 64-bit.

For your second question, if you extract a particular slice ( $z = k$ ) on the 3rd dimension, that slice index will be moved from k to 1 ( $z = 1$ ). Since affine transformation matrix uses the x, y, and z indices for its input, the image transformation will no longer be able to carry out if you only read a particular range of x, y, or z. Therefore, "load\_nii" only allows a subset data retrieval for the 4th dimension and up.

---

## **NIfTI data dimension related question**

I need to collapse a time series of 3D nii images which all have the same voxel intensities, but one of them has a different scaling factor (generated by SPM ImCalc). When I collapse the 3D images together with 'collapse\_nii\_scan', the script imposes to this single different scan the same scaling factor as for the other images in the series, but without preserving image intensity, so that this single scan ends up with a different image intensity than the other images in the series. Is there any modifications that could be made to the 'collapse\_nii\_scan' script, in order to preserve intensities? (31-January-2012, Extracted from Marco Tettamanti at Nuclear Medicine Department & Division of Neuroscience, San Raffaele Scientific Institute, Italy)

## **Answer**

Before presenting you a solution, I need to address 2 points here:

1. "collapse\_nii\_scan / expand\_nii\_scan" are designed to process a run of homogeneous MRI scans. i.e. all the information in the header are supposed to be exactly the same. So it presumes I can use any one of the header info for the collapsed header (I used the first one).
2. Because "\_untouch\_" are used in the programs, when you collapse 3D images together, the script actually does not change any of the image intensity. i.e. the collapsed image use the first header info, and all image intensity are preserved. The reason that you found the image intensity is changed is because the header for that particular scan is different from the first header info.

In order to collapse the images together, you need to have the same volume dimension, same voxel size, and same originator. So you have all this. In order to accommodate the singular image, you will have to interpret all the header info. The easiest way to do this is to follow the steps below (assume that the singular image is not the first image):

1. Collapse the images just as you did. e.g. `collapse_nii_scan('* .img')` or `collapse_nii_scan('* .nii')`
2. Load the singular image without "\_untouch\_" (with touch). e.g. `nii1=load_nii('special_image.img')` or `nii1=load_nii('special_image.nii')`
3. Load the collapsed image without "\_untouch\_" (with touch). e.g. `nii=load_nii('multi_scan.img')` or `nii=load_nii('multi_scan.nii')`
4. Using: `view(nii)` to identify the singular scan ID. you can enter the ID number or using the scroll bar below "scan ID".
5. Replace the singular scan. Let's say it is the 2nd one, then this will replace the singular scan: `nii.img(:,:,2)=nii1.img;`
6. After replacing, please use `view(nii)` again to make sure that everything is fine.
7. If everything okay, you can use `save_nii` to save the 'nii' to a new file name.

---

## Parameter scl\_slope related question

I am trying to change the pixel value 1 into 5 and then save it as .img. But if i open these images in viewer like MRICroN and FSLVIEW it shows a pixel value as 0.01961. I could not find the reason. Can you please help me in finding the solution to the above problem. I have put the lines of code which were used by me: `[nii1.hdr, nii1.filetype, nii1.fileprefix, nii1.machine] = load_nii_hdr('1sub_110_A-128_5med');`  
`[nii1.img, nii1.hdr] = load_nii_img(nii1.hdr, nii1.filetype, nii1.fileprefix, nii1.machine); z=find(nii1.img); %number of index---4610`  
`img1=zeros(size(nii1.img)); img1=uint8(img1); img1(z)=5.00000; nii1.img=img1; save_nii(nii1,'out1.img');` Thanks in advance for your help.  
(31-August-2011, Extracted from Priya Lakshmi from priyalakshmi79@gmail.com)

## Answer

You are not supposed to use "load\_nii\_hdr.m" & "load\_nii\_img.m". Both are internal functions, and are not supported when you call them directly.

If you want to take look at the header information, please use "load\_untouch\_header\_only.m".

If you would like to change the voxel value, please use "load\_nii.m" to load the NIfTI file, and then use "view\_nii.m" to edit voxel value at crosshair (under "Edit" tab), and then click "Save displayed image" under "File" tab.

The reason of inconsistency in your case is caused by the scaling factor in the header that was not properly interpreted by yourself.

---

## Parameter scl\_slope related question

I have been using your load\_untouch\_nii and save\_untouch\_nii to do CBF analysis. However, I just noticed that my out files have data ranges much higher than the matrix I saved. I was wondering if during the saving, you may change anything in terms of scaling, or data type? Attached please find the script. (22- February-2010, Extracted from Najmeh Khalili-Mahani at Montreal Neurological Institute, McGill University, Canada)

## Answer

This is because you were using "save\_untouch\_nii" to save the "DATA", but using "load\_nii" to load to "DATA2".

In "DATA.hdr.dime" structure, there are two fields "scl\_slope" and "scl\_inter", which controls the scaling. When you use "load\_untouch\_nii / save\_untouch\_nii" pair to DATA, none of the fields in the header will be touched, and they will not be applied to "DATA.img" either.

However, when you use "load\_nii / save\_nii" pair to DATA2, all fields in the header will be properly applied to "DATA2.img" and then the field will be changed.

---

## Parameter scl\_slope related question

I have 3D ANALYZE volumes with different scale factors and I want to store them as a 4D NIfTI file. Can your toolbox (or the NIfTI format) handle this so the different scale factors still can be available. It seems to me like the scale factor is set to zero. (19-April-2007, Extracted from Peter Mannfolk at Medical Radiation, Lund University, Sweden)

## Answer

Because my NIfTI toolbox has already taken care of the scale factor, the scale factor has to be set to zero.

When you use "load\_nii.m" to load the 3D ANALYZE volumes, you don't have to worry about the scale factor any more, because scl\_slope & scl\_inter from each different ANALYZE volumes have been applied to the corresponding img matrix, and the scale factors are all set to 0.

All you need to do is to concatenate those 3D img matrices into a 4D matrix, and run:

```
nii = make_nii(4Dmatrix, [options]);    % build 4D NIfTI structure
save_nii(nii, '4D.nii');               % store as a 4D NIfTI file.
```

---

## Parameter scl\_slope related question

There is a discrepancy to read voxel values between the two cases: (1) Values are read using combination of inbuilt functions fread() and fopen(); (2) Values are read using load\_nii(). The values read by fread() function is different from the values in "nii.img". Can you throw some light on the problem of getting too small values? (02-August-2006, Extracted from Harshal Oza from harshaloza@yahoo.com)

## Answer

You may want to look at the following URL:

[http://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields\\_pages/scl\\_slopeinter.html/view?searchterm=scl\\_slope](http://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields_pages/scl_slopeinter.html/view?searchterm=scl_slope)

Since your scl\_slope value is nonzero (9.4500e-003), each voxel value in your "nii.img" has to be scaled by:

```
y = scl_slope * x + scl_inter;
```

where, x is the stored voxel value in your image file, and y is the true voxel value in "nii.img".

You can get your original "scl\_slope" and "scl\_inter" in "nii.original.hdr.dime" structure. If you want, you can restore original "nii.img" by:

```
stored_val_img = (nii.img - scl_inter) / scl_slope;
```

---

## Parameter scl\_slope related question

I'm having some problems with the scaling factor. The value of `nii.hdr.dime.scl_slope` is zero, instead of the correct value (0.003922) that I am expecting using `spm5` or `avwhd` from FSL. (07-March-2006, Extracted from Ged Ridgway at University College London, UK)

## Answer

Actually both "scl\_slope" and "scl\_inter" parameters have been correctly read and applied to the "img" variable of the output NIfTI structure (see "xform\_nii.m" file): `nii.img = scl_slope * double(nii.img) + hdr.dime.scl_inter;`

The reason to display "hdr.dime.scl\_slope" as zero is a feature of my package. It signifies that the "img" variable has already been modified. You can also find that "hdr.hist.qform\_code" and "hdr.hist.sform\_code" will be set to zero after you use "load\_nii.m".

The original value of "hdr" can be easily found by: `hdr = load_nii_hdr(filename);` You should have correct value (0.003922) there.

---

## General question

When I tried to load my NIfTI file with "load\_nii.m" and resave it with "save\_nii.m", I ran into some trouble. The new file is quite different in the header. I just want to average some frames along the time-series, how do I save exactly as loaded? (27-August-2008, Extracted from Michael Marxen at Rotman Research Institute, Canada)

## Answer

When you use "load\_nii.m" to load a NIfTI file, the information in NIfTI header is appropriately applied, which includes affine space transformation and voxel intensity scaling etc. The NIfTI header is then changed to indicate that its information has been properly applied.

If you would like to save exactly as loaded, you have to use "load\_untouch\_nii.m" to load a NIfTI or ANALYZE file. In this case, nothing is changed. That means, you have no idea where (e.g. Left | Right) is a voxel in loaded 3D matrix without matrix computation together with the sform / qform transformation matrix in the header. Therefore, you cannot view or save the untouched data with "view\_nii.m" or "save\_nii.m" program.

If you use "save\_untouch\_nii.m" to save the data that is loaded with "load\_untouch\_nii.m", the new file is untouched. In your case, you can average frames along the 4th dimension of the 4D matrix, and adjust hdr.dime.dim(5) before save into new file with "save\_untouch\_nii.m".

---

## General question

I was wondering if I can use the command load\_nii\_hdr. I noticed that most of the header fields are same as the untouch header (when loaded with load\_untouch\_nii) and there some more fields added like flip etc. Does load\_nii\_hdr modifies any header field? or it just adds few more fields for some inner operation? (19-April-2011, Extracted from Chitresh Bhushan at University of Southern California, US)

## Answer

"load\_nii\_hdr.m" command is an internal function, and should not be called directly.

"load\_untouch\_header\_only.m" command is available to get the header fields without any modification. It loads exactly whatever stored in the header section of NIfTI or ANALYZE data, and can automatically detect the input file format. NIfTI structure will be returned for NIfTI file, and ANALYZE structure will be returned for ANALYZE file.

---

## General question

Can I run this package under MATLAB R2008a? (18-June-2008, Extracted from Holger Krause at Neurology/MEG, Heinrich-Heine University Düsseldorf, Germany)

## Answer

Yes, of course. You can run any programs in this package under any version of MATLAB on any platform. However, the speed of "reslice\_nii.m" program would be much slower if you run it under MATLAB 6.5 (R13) or below. This is because MATLAB overhauled its computation kernel on MATLAB 6.5 (R13) and above.

---

## Question about unsupported file format

I try to read ANALYZE file in MATLAB. I am not using ".img / .hdr" or ".nii" format, but ".avw" format where the header is also embedded. I cannot read those files using your tool. Am I missing something or is this file format not supported? (12-May-2006, Extracted from Olivier Salvado at Biomedical Engineering, Case Western Reserve University, US)

## Answer

The ".avw" file format that you mentioned is only used by *Analyze software* from *AnalyzeDirect Inc.* It is not the old ANALYZE 7.5 file format that people usually referred to, and ".avw" file format is not compatible with NIFTI file format at all. So it is not supported by my tools.

The ".img / .hdr" file format is used by both NIFTI file format and old ANALYZE 7.5 file format, while the ".nii" file format is used by NIFTI only. NIFTI file format inherited old ANALYZE 7.5 file format, and added much more new features. e.g. it can include time series into the file (4-D file) etc. Therefore, old ANALYZE 7.5 file format is compatible with NIFTI, and both are supported by my tools.

If your .avw file is in non-compressed format, thanks to Jeff Gunter from Mayo Clinic, you can use [loadavw.m](#) to load your .avw file into avw structure in MATLAB. Since the data sits in avw.Image, you can then use my "make\_nii.m" to convert it to NIFTI structure.

If your .avw file is in compressed format, you will have to load it back with that *Analyze software*, and then save as *AnalyzeImage7.5* file format again, which can be loaded by my NIfTI tools directly.

---

## Question about unsupported file format

I'm planning on using your tools to convert some binary analyze 7.5 files into nifti format. While I was looking around I thought I'd ask if you happen to know anyway to convert an analyze .obj file straight to nifti. We are currently going from .obj to analyze 7.5 binary and then converting to nifti. So a shortcut would be nice. (15-February-2012, Extracted from Ryan Johnson at MIND Institute, UC Davis, US)

## Answer

I guess that you mix Mayo's Analyze 7.5 format with *Analyze software* from *AnalyzeDirect Inc.*

My toolbox can read / write both Mayo's Analyze 7.5 format and NIH's NIfTI format, since the latter one is a variation of the former one. However, my toolbox is not designed to convert from one format to the other.

The ".avw", ".obj", ... etc. are result files from AnalyzeDirect Inc. My toolbox can never read those files. The only suggestion that I can give you for these type of files is to let Analyze software from AnalyzeDirect Inc convert it to either Mayo's Analyze 7.5 format or NIH's NIfTI format.

---