

The Slice Noise Scripts

Antonia Hamilton, Dartmouth College, 2007
antonia.hamilton@dartmouth.edu

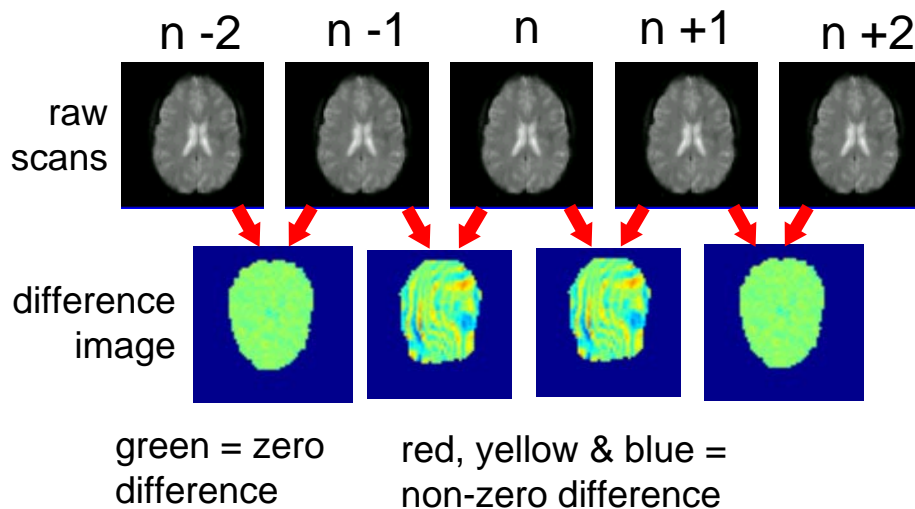
The aim of these scripts is to locate, quantify and fix the scanner artefacts which have been observed in some EPI sequences collected on the current Dartmouth 3T scanner.

This documentation begins with a brief description of the sources of noise and the means of locating them. Then I provide a description of each of the slice_ functions and how to use them.

Introduction to sources of noise

A basic assumption in fMRI data is that the data is temporally smooth. That means that scan **n** should look very similar to scan **n+1**, and slice **i** out of scan **n** should look very similar to slice **i** out of scan **n+1**. We can use this assumption to check the integrity of the raw data from the scanner. If a slice does not look like the equivalent slice in the next scan, something is wrong.

In the set below, scan **n** is unlike its neighbours, so this scan is noisy.



There are 2 major reasons why a slice might not look like its neighbours

Movement – if the participant moved in the scanner, even less than one millimetre, the slices may be different

Scanner artefact – if slice data was not collected correctly by the scanner, the slice will not look like its neighbours

Unfortunately, there is no automatic way to distinguish between head movement and scanner artefact in brain images. So if you want to remove artefact slices from an EPI sequence, you need to learn what slices look like with movement and artefact, and visually inspect each slice to decide which to reject. The `slice_` scripts aim to help you do this as efficiently as possible.

There are 6 scripts. The following pages explain how to use each script, what the script does and why.

IMPORTANT NOTE

The `slices_` scripts allow you to detect and replace slices which have scanner artefact. Bad slices will be replaced with the average of their neighbours. This operation will alter the results you get in your statistical analysis, and may not be statistically valid. It is up to you to decide how many slices you are prepared to replace in a session, or whether a whole session has too much artefact and cannot be used.

Setting up the slice_scripts

1. Copy the whole of /afs/dbic.dartmouth.edu/DBIC/scratch/slice_scripts including the noise_templates to your local directory on AFS
2. Edit **slices_defaults.m** in your favourite text editor as shown below
3. Run **slices_defaults.m** to save a new defaults file

NB. These scripts have only been tested under AFS with spm2. They probably won't work under Windows

Enter your AFS path here so that the script can find /noise_templates in your local folder

And again here

If you want to change the default thresholds, edit here

```
clear all, close all
disp(' ')
disp('This is the slices defaults script')
disp('You will want to edit this in your favourite text editor')
disp('to set the right paths and thresholds')

*** Edit the next 2 lines to change the EPI template location
sdef.epi_template = ['/afs/dbic.dartmouth.edu/usr/grafon/antonia/' ...
                    'artifact_tools/noise_templates/xEPI.img'];

disp(' ')
disp('The current EPI template location is:')
disp(sdef.epi_template)
if(exist(sdef.epi_template)==2)
    disp('This file is OK')
else
    disp('Cannot find the file, please edit slices_defaults.m')
end

** edit the next 2 lines to change the brain template location
sdef.brain_template = ['/afs/dbic.dartmouth.edu/usr/grafon/antonia/' ...
                      'artifact_tools/noise_templates/xwhole_brain.img'];

disp(' ')
disp('The whole brain template location is:')
disp(sdef.brain_template)
if(exist(sdef.brain_template)==2)
    disp('This file is OK')
else
    disp('Cannot find the file, please edit slices_defaults.m')
end

sdef.th = 25;    ** edit this number to change your threshold
disp(' ')
disp(['The current noise threshold is: ', num2str(sdef.th)])
```

Slices_analyse.m

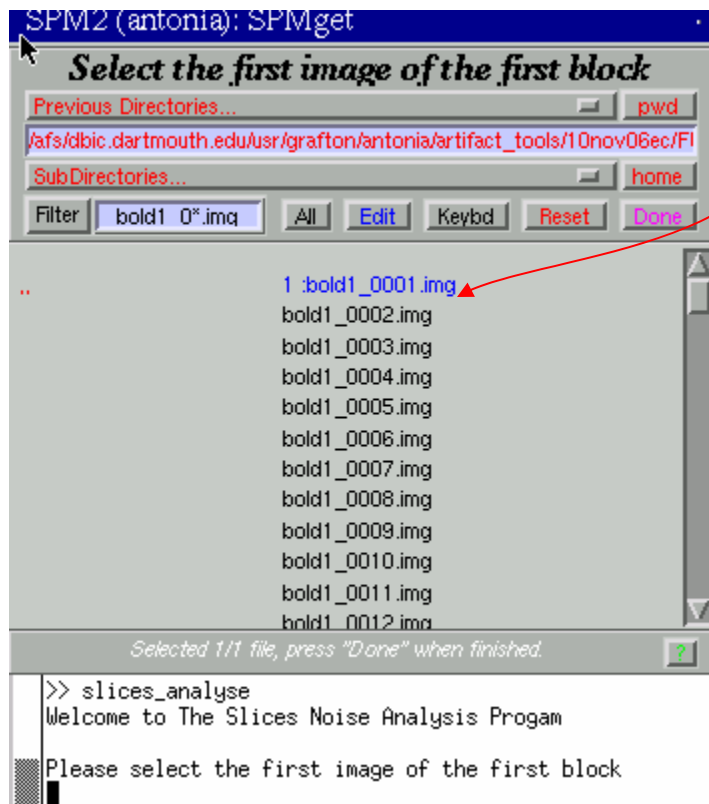
This is the first script to run. It gives you an overview of the number of bad slices in each scanning session. You need to run the script once for each participant, it can do multiple sessions in one go.

To run –

Type **slices_analyse** at the command line

Follow the instructions

- enter the first image of the first session
- enter the number of sessions
- enter a filename for saving the results



How it works -

This script works in several stages

- realign all the images with `spm_realign` to get movement parameters
- reslice to get a mean EPI image
- normalise the EPI template to the mean EPI image
- apply the normalisation to a mask of the brain. This gives you a brain mask which is specific to this person and this session
- Looking only within the brain mask, calculate the noise on each slice as

$$\text{Noise}_x = \frac{\sum_{n=-5}^{n=+5} (\text{Slice}_{x-n} - \text{Slice}_x)}{10}$$

It is very important to remember that noise is a DIFFERENCE measure. The noise in an image is affected by the current image, but also by the neighbours you are comparing it to. If the neighbouring images are noisy or have movement, the current image may look noisy too.

Noise is measured in arbitrary units. Good slices have a noise value of 10-20. The default threshold for deciding if a slice might be bad is 25. The worst slices have a noise level of about 80 units.

The output of the script is a nice blue image showing the overall pattern of noise (next page) which is saved in the file name you entered at the start. You will need to select this file as the input in each of the next stages.

Slices_summary.m

This script plots out a nice summary page for each session.

It is called by most of the other scripts, but you can also run it from the command line by typing **slices_summary**. You will be prompted for a saved noise file (from slices_analyse).

Overall noise image.

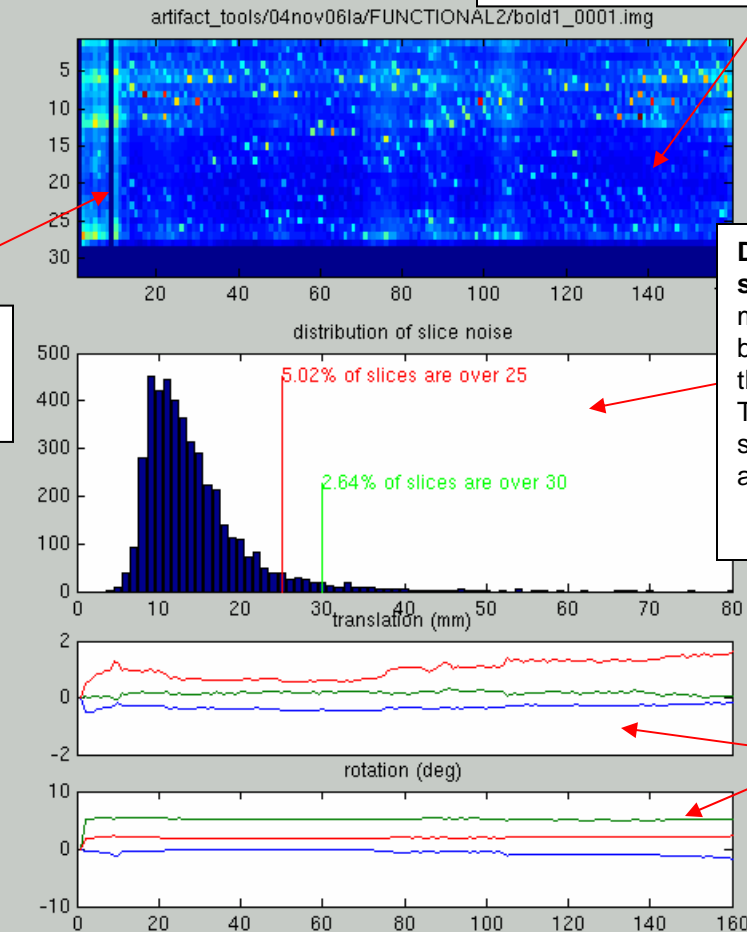
X axis = scans, Y axis = slices

Very dark blue regions have no data (because of too much movement or empty slices).

Mid blue regions indicate good data

Pale blue, yellow and red spots indicate noisy images. Noise sprinkled like confetti is mainly due to artefact. Noise in wide vertical bands is mainly due to head movement.

Dark blue = no data because of head movement



Distribution of noise over slices. In a typical session, most slices have low noise but some are above the thresholds (red & green). Typically, 50% of noisy slices are artefacts and 50% are movement.

Movement parameters for this session

Slices_examine.m

This script shows you single slice images from a sample of your dataset. It is useful for learning about what artefacts and head movements look like, and for broadly exploring your data.

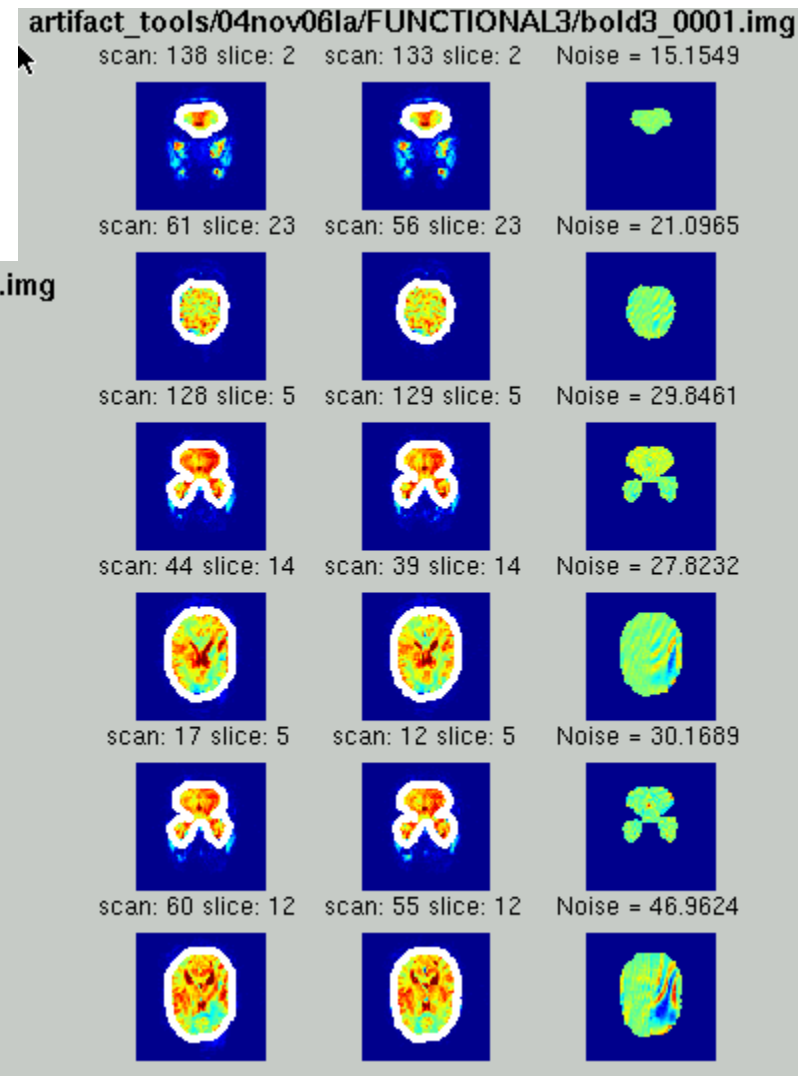
It doesn't make any changes to the data, so once you are familiar with what artefact and noise look like, you can skip this step.

Example slice

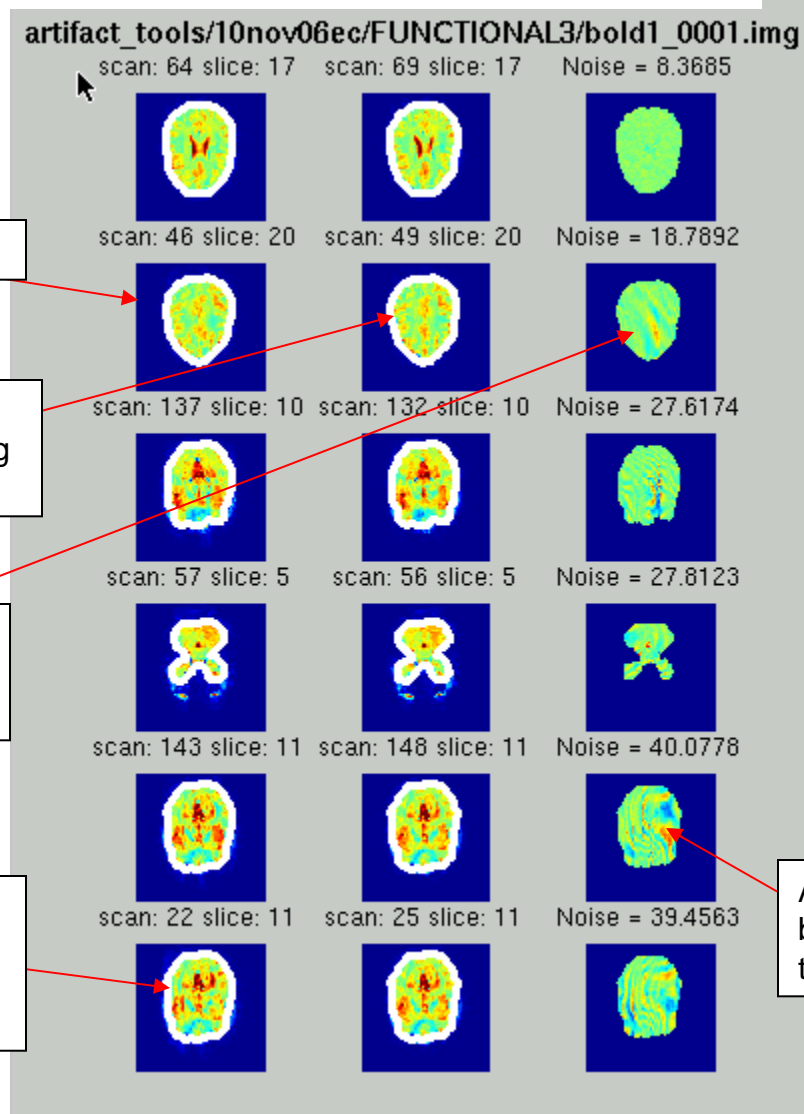
Comparison slice from a neighbouring scan

Difference image showing a little noise

Fat white border indicates the edge of the brain mask on this slice



Artefact looks like red and blue swirls or fingerprints in the difference images



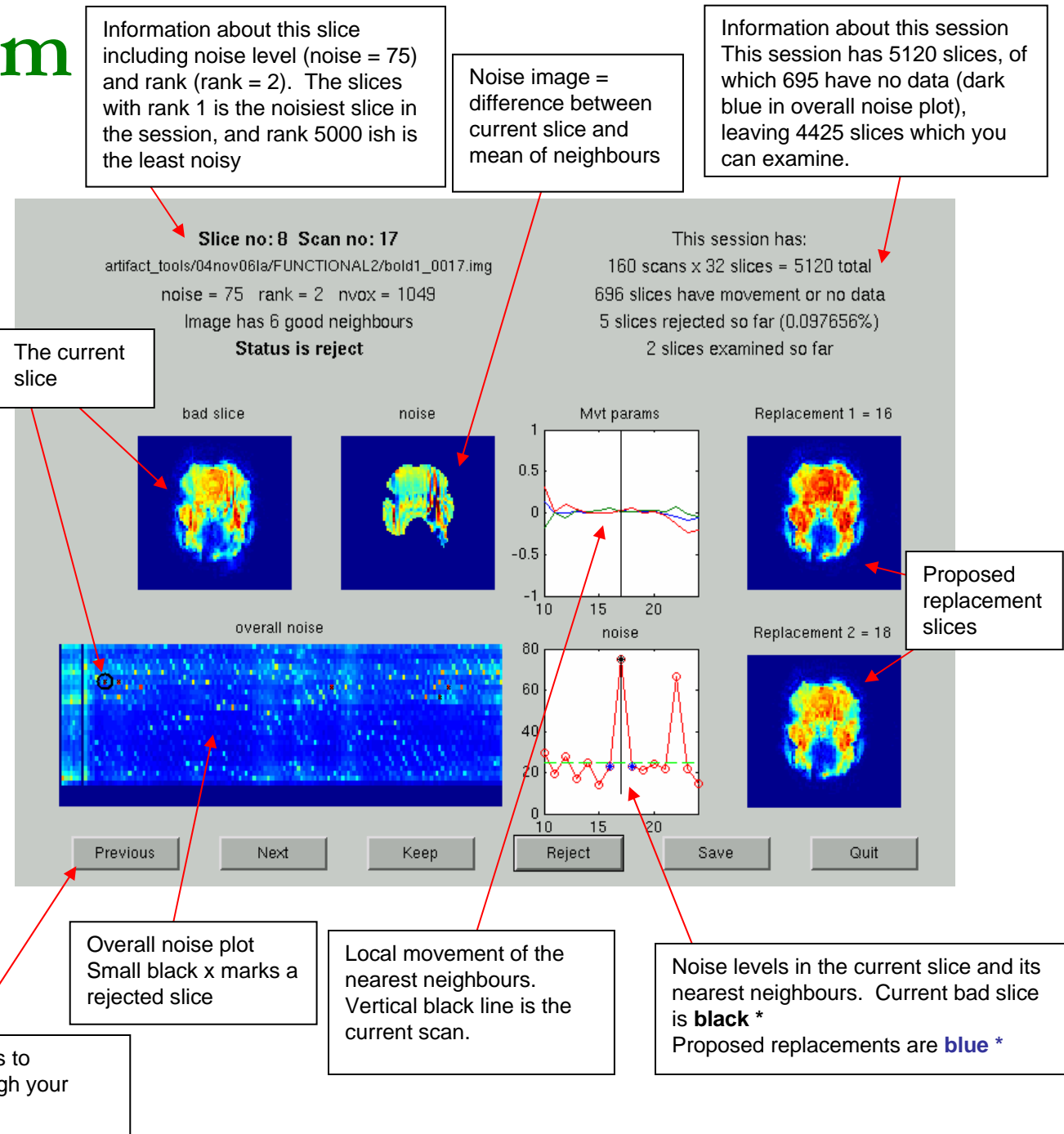
Slices_reject.m

This script lets you examine each slice in detail and decide which ones to reject. By default, all slices are accepted. You have to click the 'reject' button to reject a slice.

The script will show you every slice of your data, starting with the worst (greatest noise) and continuing in order of noise. You should probably examine slices manually until you get to a point where you find you are accepting every slice (roughly, noise < 25). Then you can press SAVE and the rest of the slices will be accepted by default. This means you only have to visually examine about 200 out of your 5000 slices.

At this stage, none of the images are actually edited. You are just creating a list of the images you want to reject in the next step. So you can run this script as many times as you like without damaging your data.

For every slice, you will see an image like this:

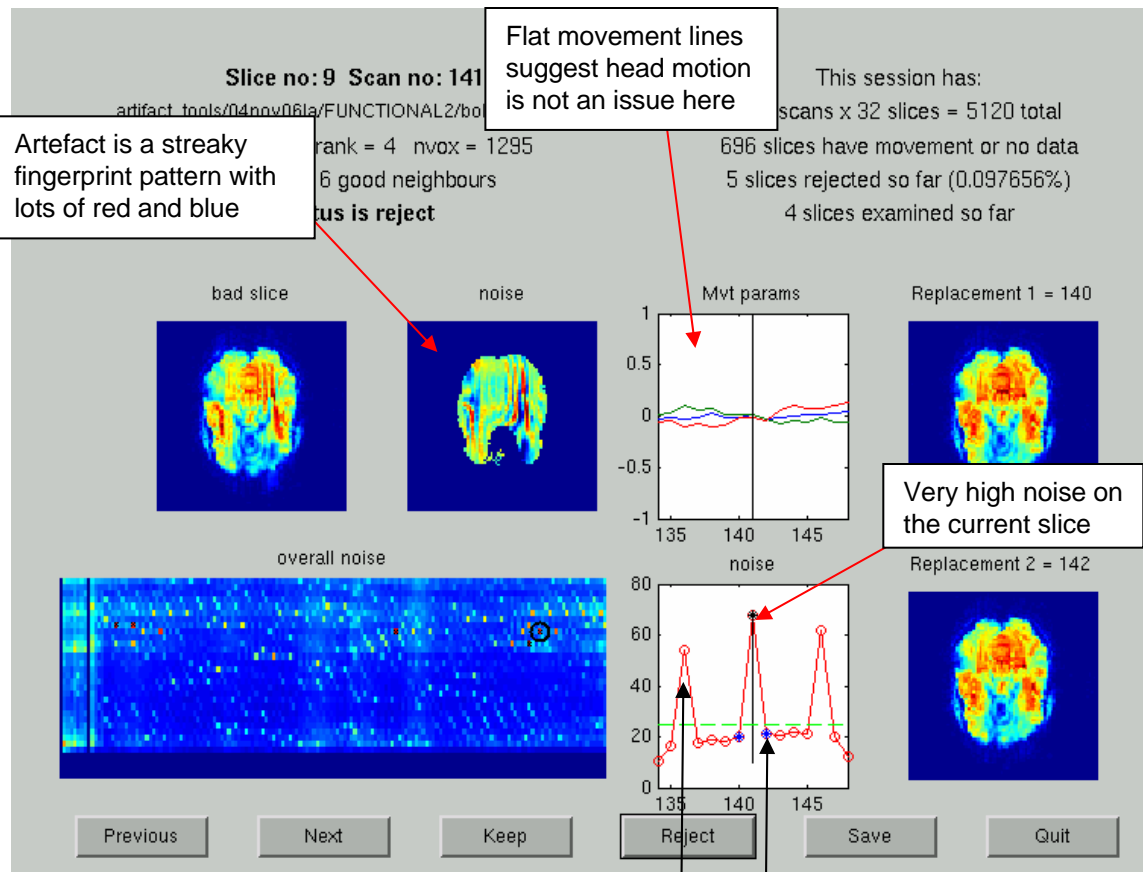


Slices_reject.m

Here is another example slice with the noise annotated

These two examples are slices you should REJECT. They have high noise (> 25 units) and very little movement. Click the Reject button to mark a slice for rejection. A little black X should appear on the slice in the overall noise plot.

If you make a mistake in clicking the buttons, you can use 'Previous' and 'Next' to move back and forward in the list of slices without changing the status of a slice. Then you can press 'keep' or 'reject' again if you like. Press 'Save' to save you current list of rejected slices and 'Quit' to exit. When you start the script up again, you can start from where you left off last time if you want.



Slices_reject.m

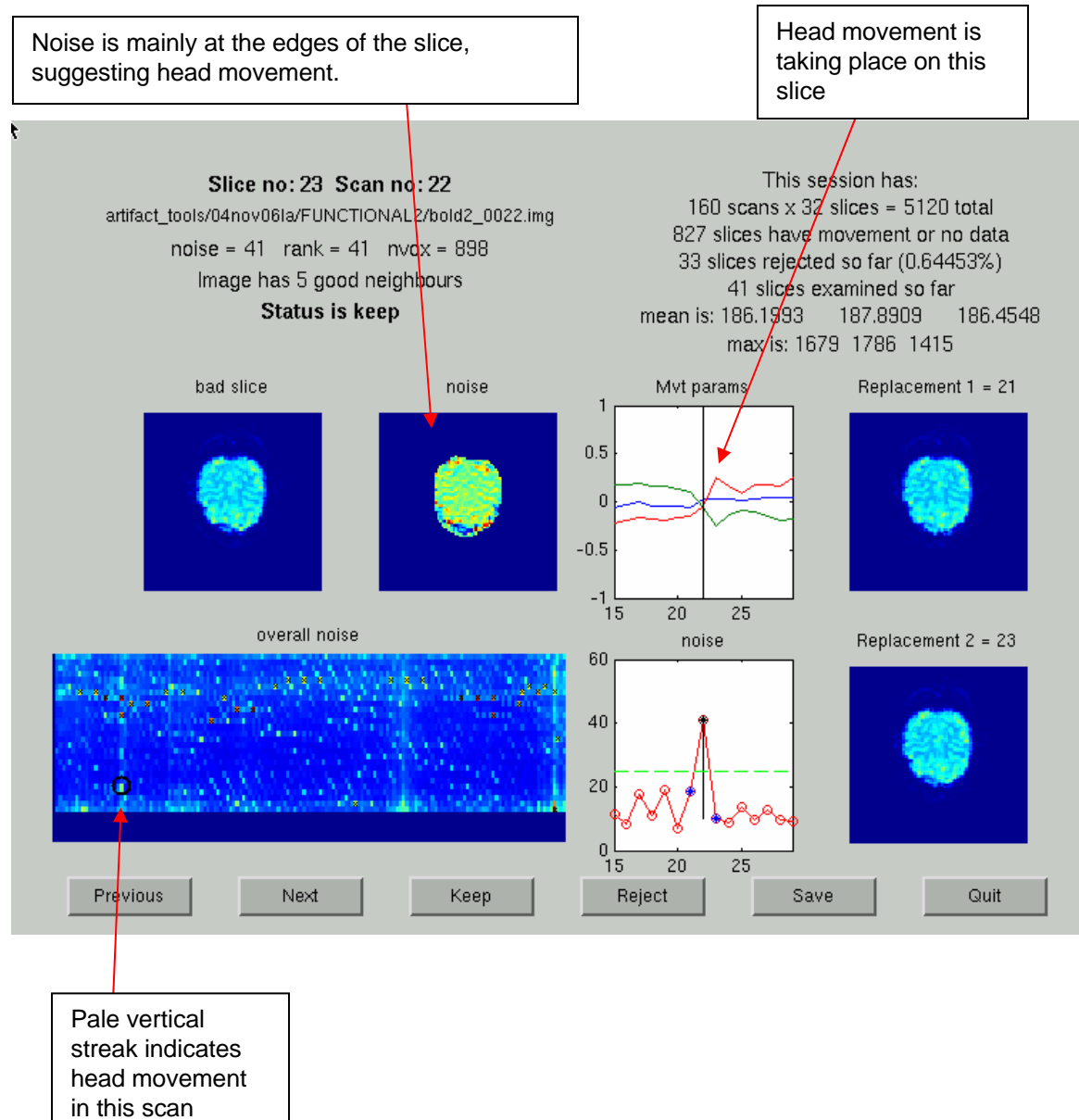
Some slices may be noisy because the person's head moved, as in this example

You should press 'Keep' for this slice.

Judging which slices are artefact and which are motion is not always clear cut. It is probably best to be conservative and keep a slice if you aren't sure, especially if there aren't good possible replacement slices among the immediate neighbours

The script warns you if the closest neighbours are above threshold or don't exist.

Then you should probably keep the slice



Slices_reject.m

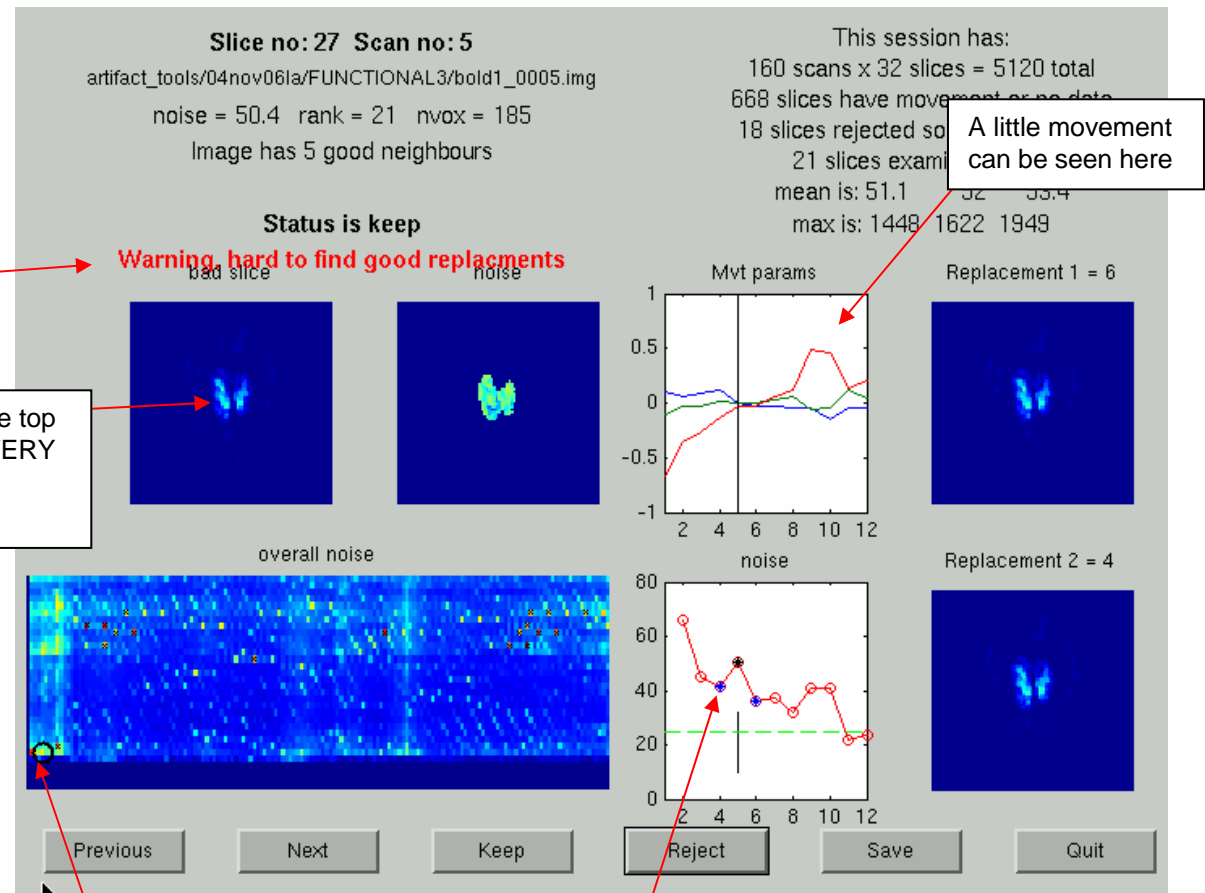
This example is a close call, but I would probably keep the slice

Script warns that possible replacements are also noisy

Small slices at the top of the brain are VERY susceptible to movement

This slice comes from part of the session with some movement

Possible replacements also have noise, because they are also small slices.



A little movement can be seen here

Slices_replace.m

This script takes the list of bad slices (from slices_reject) and replaces those with the replacement slices, as pictured in the slices_reject script.

First you get a plot like this confirming which slices will be replaced

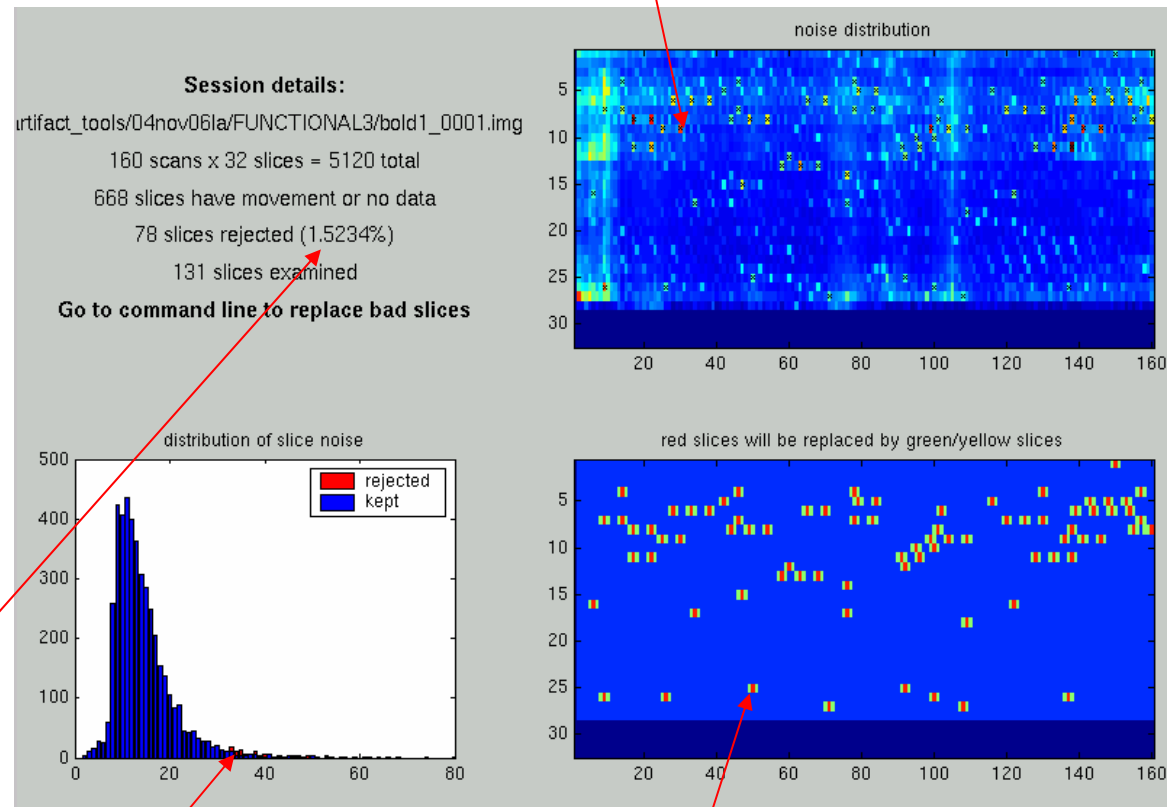
Then if you select Y at the command line, the script will read in the bad images and replace the slices with slices from neighbouring images.

After the slices_replace script has run, you may want to run slices_analyse again to see the difference

1.5% of slices will be replaced in this session

Little red bars are the slices which will be replaced.

black Xs mark the slices to be replaced



Each red slice will be replaced with the average of the green slices next to it

Before - After

1.5% of slices were replaced in this test subject. The image on the right is cleaner and has fewer slices over threshold.

More slices could be replaced, but that would start to invalidate the statistical analysis.

This particular subject should probably be rejected altogether.

