```
GNERAL UNIX INFORMATION:

$> stands for the Unix command prompt (which may vary from user to user).
The text typed after the prompt is the command with various options.
In general, in Unix, the order of events in a command request is
(i.e. the command syntax):

   a       b     c
command <options> file


               A Unix Crib-sheet: (By Kathleen Frederickson)

(Let's imagine a sample pathname of neuro/data/howard/aging/brain
and another one of neuro/pet/christine/analysis/controls)

cd means "change directory" -- in a variety of permutations, it allows
you to move through the many branches of stored information

cd ..   will take you up one level on the path in which you find
yourself (e.g. you're in neuro/data/howard/aging, you use this
command and you'll be in neuro/data/howard)

cd ../../../ will take you up three levels (another ../ would take you
up four, one less would only take you up two and so forth).  So if
you're in neuro/data/howard/aging/brain and you type "cd
../../../" and you'll be in neuro/data/

cd [subdir] if you want to go to the next subdirectory down the path
(e.g. you're in neuro/data/ and want to be in neuro/data/howard) you
have only to type cd and then the name of the subdirectory (e.g. cd
howard) WITHOUT A SLASH

cd /[dir] if you want to go to another directory altogether
(e.g. you're in neuro/data/howard and you want to be in
neuro/pet/christine) you type cd and the FULL name of the directory
WITH a slash at the beginning (e.g. cd /neuro/pet/christine).This will
also work to move you back within a given string (e.g. from
/neuro/data/howard/aging to neuro/data) but you need to type out
the WHOLE string, right from the beginning (e.g.cd /neuro/data)

CTRL-C CANCEL.  A very important and very useful command.  If the
computer is still processing an operation, Control-C will stop it
mid-stream.  Moreover if you've typed a line but haven't hit enter, it
will delete your typing and move you back to the cursor prompt.

pwd will tell you where you are within the whole structure of
directories,subdirectories and so forth.  Type it and something to the
effect of "/neuro/data/howard/" will pop up.

mkdir [name] will create a directory WITHIN the directory in which you
are working. For example, you're in /neuro/data/ and you type "mkdir
kathleen" you will have created /neuro/data/kathleen/

ls will list the files and subdirectories within a the directory where
you are working

ls -l same as the "ls" command, but this will ALSO tell you what the
rights (e.g.who can read, write and execute the file) are for each
file and it will also provide information on file size and last
date/time it was modified.

df -k will tell you how much space is left on the assorted disks
attached to the system.  df -k . will tell you how much space you have
on the current disk.

cp              COPY
mv              MOVE
rm              REMOVE

There are two procedures for all of these commands: one for copying,
```



```
moving and removing FILES; and one for copying, moving and removing
DIRECTORIES.

FILES: to copy a file you type "cp [filename] [directory into which
you want to copy the file WITH SLASH]"--so, let's say you've got a
file called "smith" located in "/neuro/data/howard" and you want to
copy it to "/neuro/pet/christine/" IF YOU ARE ALREADY IN
"/neuro/data/howard"you need only type "cp smith
/neuro/pet/christine";IF YOU ARE NOT,you need to type "cp
/neuro/data/smith /neuro/pet/christine

If you substitute "mv" for "cp" you will move the file instead of
copying it.

If you want to remove a file you just need to type the first half
(and substitute "rm" for "cp" (e.g. to remove "smith" you'd type "rm
smith"from WITHIN /neuro/data/howard or "rm /neuro/data/howard/smith"
if you are elsewhere

DIRECTORY: to copy a directory you need to add an extra "-r" after
"cp".So, you want to copy the "howard" directory you type "cp -r
/neuro/data/howard/ /neuro/pet/christine".  You also need to add the
"-r" for moving and removing directories....apart from the "-r" the
procedure is the same.

less [filename]: will scroll through a file, one screenful at a time
(e.g. you want to look at file smith, which is REALLY big, you type
"less smith" and it will show it to you).  If the file you're looking
at is in ASCII format, you will only see the text contents, if it's a
binary file (e.g. a Word file, or WordPerfect file) the system will
warn you that it may be binary and ask you if you want to continue.
If you say yes, the system will show you not only the contents of the
file, but also the machine code et. al.


DATA MANAGEMENT: (by Kate Hanratty)

After PET and MRI studies have been completed, gather the image data
and organize it in such a way that the "automagic" processes can run
properly.

In particular, you will want to run the

1. automated MRI image registration do_mritotal (generates a transform
   to a standardized coordinate system, known as stereotactic space
   and corrects for motion artifacts in the MRI volume) and

2. automated PET-MRI image registration (generates a transform to map
   the MRI volume to a 'baseline' PET volume)

3. automated PET-PET image registration (generates a transform to map
   the subsequent PET volumes to a 'baseline' PET volume to correct
   for inter-scan & intra-scan motion artifacts)


To collect the image data, you will need a list of subjects, dates and
times of PET and MRI scans and the start time for each PET
scan/condition.  It is a good idea to copy the PET information from
the PET log book kept by the PET technicians.

Data management operations (i.e., finding files, copying files, etc.)
should be executed from the main processing computer at the BIC:
bottom.

To make your life easier and to allow the automated image processing
programs to run effectively, please follow these directory and file
naming conventions:

In your top-level working directory (also called the "base"
directory) e.g. /data/aging/nling/mystudy5_dat, make two
separate sub-directories for the PET and MRI data:
```

```
/data/aging/nling/mystudy5_dat> mkdir pet
/data/aging/nling/mystudy5_dat> mkdir mri

Copy only the "raw" PET data for your DAT subjects to the pet
directory and the only the "raw" MRI data to the mri directory.  Other
directories will be created by the system to store information
generated by these pre-processing steps.


PET DATA:

To find the archived PET data, use the following command: (in this
example, you only want to find the image information for the AD patients
and you want to copy or write the results of this search to a file that
will be saved in the /data/aging/nling/docs directory in a file
called mystudy5_dat_petfiles.txt)

$> list_ecat_archive -image -l name1 name2 ... name55

[ KATE NEEDS TO CLEAN UP THIS SECTION!! ]

You can specify as many names as you want, either first name or last
name, separated by a space. Here is a sample search for three patients
(bento, hierophant and bubble):

$>list_ecat_archive -image -l bento hierophant bubble > /data/aging/nling/docs/s
ample_petfiles.txt

File type  OD    Size    Date        Time     File Name
-------------------------------------------------------------------------------
ECAT IMAGE 36A   2065920 19 Nov 1998 11:18:15 Bento_Phil_85f_4875_de8.v
ECAT IMAGE 36A   2065920 19 Nov 1998 11:31:47 Bento_Phil_85f_4878_de11.v
ECAT IMAGE 36A   2065920 19 Nov 1998 11:47:04 Bento_Phil_85f_487b_de14.v
ECAT IMAGE 36A   2065920 19 Nov 1998 12:00:57 Bento_Phil_85f_487e_de17.v
ECAT IMAGE 36A   2065920 19 Nov 1998 12:11:14 Bento_Phil_85f_4881_de20.v
ECAT IMAGE 36A   2065920 19 Nov 1998 12:24:58 Bento_Phil_85f_4884_de23.v
ECAT IMAGE 37A   2065920  7 Dec 1998 11:05:02 Bubble_Aurora_101_4a2e_de8.v
ECAT IMAGE 37A   2065920  7 Dec 1998 11:14:34 Bubble_Aurora_101_4a31_de11.v
ECAT IMAGE 37A   2065920  7 Dec 1998 11:32:26 Bubble_Aurora_101_4a34_de14.v
ECAT IMAGE 37A   2065920  7 Dec 1998 11:42:47 Bubble_Aurora_101_4a37_de17.v
ECAT IMAGE 37A   2065920  7 Dec 1998 11:52:00 Bubble_Aurora_101_4a3a_de20.v
ECAT IMAGE 37A   2065920  7 Dec 1998 12:01:40 Bubble_Aurora_101_4a3d_de23.v
ECAT IMAGE 36B   2065920  2 Dec 1998 13:51:35 Hierophant_Sylvana_87d_49c0_de8.v
ECAT IMAGE 36B   2065920  2 Dec 1998 14:03:24 Hierophant_Sylvana_87d_49c3_de11.v
ECAT IMAGE 36B   2065920  2 Dec 1998 14:18:05 Hierophant_Sylvana_87d_49c6_de14.v
ECAT IMAGE 36B   2065920  2 Dec 1998 14:28:53 Hierophant_Sylvana_87d_49c9_de17.v
ECAT IMAGE 36B   2065920  2 Dec 1998 14:39:58 Hierophant_Sylvana_87d_49cc_de20.v
ECAT IMAGE 36B   2065920  2 Dec 1998 14:50:59 Hierophant_Sylvana_87d_49cf_de23.v

OD stands for optical disk - all PET data is archived to optical disks
numbered 1 to whatever (by December of 1998, there were 37 disks).  It
also indicates whether the file is on side A or side B of the disk.
This is important information!

The > symbol in the command line tells the computer to "write" the
results of the search to a file.  It is called the redirect symbol in
the Unix world.  Think of it as an arrow pointing toward the file
where you want the information stored.

Now, you can print this file so that you have a written record to work
with:

$> a2ps /data/aging/nling/docs/sample_petfiles.txt | lp

Or, if you're already in the /data/aging/nling/docs directory,
simply:

$> a2ps sample_petfiles.txt | lp

This tells the system to take the file "sample_petfiles.txt,"
```



```
convert it from ascii to postscript format and feed the postscript
information to the printer.  a2ps is the program that converts ascii
to postscript, the "|" symbol is called a "pipe" and will take the
results of the first operation (a2ps) and feed it to the next one,
which in this case is lp (the command to send a file or some
information to the printer).

If you want to save a copy of this prettily-formatted information, you
have to break the procedure into two steps:

$> a2ps sample_petfiles.txt -o sample_petfiles.ps
$> lp sample_petfiles.ps

-- End of a2ps digression --

Now, that you know on which optical disks this information is stored,
you can retrieve it, convert it to the minc format and store it in the
directory in which you will be working.

A note about optical disks used at the BIC:

PET data is stored on high density (1024 bytes per sector) optical
disks and used *only* in the optical disk drive on the Sun computer
attached to the PET scanner (known as the ECAT HR+).  MRI data is
stored on low density (512 bytes per sector) optical disks and used
only in the optical disk drive on the SGI computer known as Bottom.
Have someone like the BIC system administrator (presently
Jean-Francois Malouin) or one of the PET analysis staff (Sylvain Milot
and Peter Neelin, as of this date, March 11, 1999) show you where the
optical disks and OD drives for each of these computers are located
before you start loading things up.

To restore PET files from the archive optical disk: (Let us use the
example of Phil Bento whose information is stored on ECAT IMAGE 36A)

1) Load the disk labelled ECAT IMAGE 36A into the drive on the ECAT
HR+ computer

2) Type this command to begin the restore process:

$> list_ecat_archive -image -restore bento

The system prompts you to load the optical disk and hit enter when
ready to proceed.  It then asks you to confirm "restore" or not.
The default answer is yes, so you can press return and it will restore
the file.  If you don't wish to restore a file, for any reason, type n
or no and then hit enter and it will skip that file and go on to the
next file associated with subject Bento.  FYI, we do not normally use
the transmission images in our analysis, so you can reply "no" when
prompted to restore a subject's transmission scan (will end in
_tx#.v).

Here's what happened when I told the system to restore Bento's files:

$> list_ecat_archive -image -restore bento
Bolus Water #1 ABPAT 19 Nov 1998 11:18:15 Bento_Phil_85f_4875_de8.v
Bolus Water #2 INST  19 Nov 1998 11:31:47 Bento_Phil_85f_4878_de11.v
Bolus Water #3 HTPICN 19 Nov 1998 11:47:04 Bento_Phil_85f_487b_de14.v
Bolus Water #4 HAPICN 19 Nov 1998 12:00:57 Bento_Phil_85f_487e_de17.v
Bolus Water #5 ETPICN 19 Nov 1998 12:11:14 Bento_Phil_85f_4881_de20.v
Bolus Water #6 EAPICN 19 Nov 1998 12:24:58 Bento_Phil_85f_4884_de23.v
Restore Bento_Phil_85f_4875_de8.v? (yes/no)[yes] :
Restore Bento_Phil_85f_4878_de11.v? (yes/no)[yes] :
Restore Bento_Phil_85f_487b_de14.v? (yes/no)[yes] :
Restore Bento_Phil_85f_487e_de17.v? (yes/no)[yes] :
Restore Bento_Phil_85f_4881_de20.v? (yes/no)[yes] :
Restore Bento_Phil_85f_4884_de23.v? (yes/no)[yes] :
Doing remote restore on othello
Looking for files on optical "ECAT IMAGE 36A"
Press return when optical "ECAT IMAGE 36A" is in drive:
Looking for files on optical "ECAT IMAGE 36A"
```

```
Mounting optical
Mounted optical with label "ECAT IMAGE 36A"
Restoring file Bento_Phil_85f_4875_de8.v to /data/ecat/images2
Restoring file Bento_Phil_85f_4878_de11.v to /data/ecat/images2
Restoring file Bento_Phil_85f_487b_de14.v to /data/ecat/images2
Restoring file Bento_Phil_85f_487e_de17.v to /data/ecat/images2
Restoring file Bento_Phil_85f_4881_de20.v to /data/ecat/images2
Restoring file Bento_Phil_85f_4884_de23.v to /data/ecat/images2
Unmounting optical
$>
```

Note that the system tells you where it's putting the files: in the directory /data/ecat/images2. Now, I want to convert the *.v files into the minc format and put the minc files in my pet directory /data/aging/nling/pet, so I type this command: (and get the following *long* message from the system - it's just showing me what it's doing, I don't really need to do anything with this data)

NOTE: At this point, the system is case-sensitive to commands! So, if your filenames have capital letters in them (e.g., subject name), you *must* use them in order for the command do_ecattominc to work.

```
$>do_ecattominc /data/ecat/images2/Bento*.v /data/aging/nling/pet
processing: ecattominc -short -clobber  /data/ecat/images2/Bento_Phil_85f_4875_d
e8.v /data/aging/nling/pet/Bento_Phil_85f_4875_de8.mnc
Reading headers.
Copying frames:.Done
renamed /data/aging/nling/pet/Bento_Phil_85f_4875_de8.mnc --> /data/aging/nling/
pet/bento_phil_199811191118.mnc
processing: ecattominc -short -clobber  /data/ecat/images2/Bento_Phil_85f_4878_d
e11.v /data/aging/nling/pet/Bento_Phil_85f_4878_de11.mnc
Reading headers.
Copying frames:.Done
renamed /data/aging/nling/pet/Bento_Phil_85f_4878_de11.mnc --> /data/aging/nling
/pet/bento_phil_199811191131.mnc
processing: ecattominc -short -clobber  /data/ecat/images2/Bento_Phil_85f_487b_d
e14.v /data/aging/nling/pet/Bento_Phil_85f_487b_de14.mnc
Reading headers.
Copying frames:.Done
renamed /data/aging/nling/pet/Bento_Phil_85f_487b_de14.mnc --> /data/aging/nling
/pet/bento_phil_199811191147.mnc
processing: ecattominc -short -clobber  /data/ecat/images2/Bento_Phil_85f_487e_d
e17.v /data/aging/nling/pet/Bento_Phil_85f_487e_de17.mnc
Reading headers.
Copying frames:.Done
renamed /data/aging/nling/pet/Bento_Phil_85f_487e_de17.mnc --> /data/aging/nling
/pet/bento_phil_199811191200.mnc
processing: ecattominc -short -clobber  /data/ecat/images2/Bento_Phil_85f_4881_d
e20.v /data/aging/nling/pet/Bento_Phil_85f_4881_de20.mnc
Reading headers.
Copying frames:.Done
renamed /data/aging/nling/pet/Bento_Phil_85f_4881_de20.mnc --> /data/aging/nling
/pet/bento_phil_199811191211.mnc
processing: ecattominc -short -clobber  /data/ecat/images2/Bento_Phil_85f_4884_d
e23.v /data/aging/nling/pet/Bento_Phil_85f_4884_de23.mnc
Reading headers.
Copying frames:.Done
renamed /data/aging/nling/pet/Bento_Phil_85f_4884_de23.mnc --> /data/aging/nling
/pet/bento_phil_199811191224.mnc
```

If you list the contents of the directory, you will see that you now have six minc files for Bento:

```
/data/aging/nling/pet>ls
bento_phil_199811191118.mnc   bento_phil_199811191200.mnc
bento_phil_199811191131.mnc   bento_phil_199811191211.mnc
bento_phil_199811191147.mnc   bento_phil_199811191224.mnc
```

If, for example, you had PET information for 3 subjects on the OD ECAT IMAGE 36A, you could also have executed these commands for the group: (better to be in the /data/ecat/images2 directory for this...)


```
$> list_ecat_archive -image -restore name1 name2 name3
$> do_ecattominc Name1*.v Name2*.v Name3*.v /data/aging/nling/pet
```

Now, remove the disk ECAT IMAGE 36A from the OD drive and put it the next one so that you may get the PET data for your next subject(s). Do this for all of the optical disks on which your subject data is stored. Then, you are ready to begin working with this information!

MRI DATA: (This info. is totally out of date!! Buyer beware - mkh 02/04)

To find where the MRI data is stored, use this command: (from your working directory, e.g. /data/aging/nling/mystudy5_enc)

```
$> find_mri name1_firstname name2_firstname etc. > mrifile_location.txt
$> a2ps mrifile_location.txt | lp
```

The results of find_mri, should generate a list like this: (I want to find the location of MRI data for subjects Bento, Duke and Creveley):

```
$>find_mri bento earl creveley
"duke_tom_7777000_1_0120_mri.mnc.gz" found on mritrans_20b
"duke_tom_7777000_2_0120_mri.mnc.gz" found on mritrans_20b
"bento_phil_9139000_1_0120_mri.mnc.gz" found on mritrans_21b
"bento_phil_9139000_2_0120_mri.mnc.gz" found on mritrans_21b
"creveley_keith_8125300_1_0120_mri.mnc.gz" found on mritrans_21b
"creveley_keith_8125300_2_0120_mri.mnc.gz" found on mritrans_21b
```

If there is more than one subject with the first or last name you are searching for, you will get information on those scans too, whether they are in your study or not.

Be careful when searching for information on female subjects (who are or who have been married)!! Often, the PET and MRI units save these files using different last names. In general, the MRI unit uses a woman's original family name but the PET unit will use a woman's married name if that's the name she provides. Always be sure to have a list of married and maiden names for married female subjects.

Now, that you know on which optical disks this information is stored, you can copy it to the directory in which you will be working.

For example, we want to copy Tom Duke's MRI to our working directory /data/aging/nling/mystudy5_enc/mri. We would load the OD labelled mritrans 20 (with side b facing "up") in either optical drive 1 or optical drive 2 on Bottom. Then, from a terminal window on Bottom, type the following command:

```
$> mntopt1 r         or     $> mntopt2 r
```

The option r is used to access this information in "read-only" mode, important when using an OD for which you do not have "write" privileges or on a disk you do not wish to modify.

In general, on the MRI archives, the images of interest are located on optical1/mritrans/images (or, optical2/mritrans/images). You will need the second file acquired for a subject, the *_2_0120_mri.mnc.gz file. Use this command:

```
cp /optical2/mritrans/images/duke_tom_7777000_2_0120_mri.mnc.gz
/data/aging/nling/mystudy5_enc/mri/.
```

Where /optical2/mritrans/images is the path where the MRI file is located. And the /data/aging/nling/mystudy5_enc/mri/. indicates that the file is to have the same name in the destination directory.

Once you have copied all the files you need from the OD in the drive, you must dismount it, put it away and move on to the next one. The command to dismount the optical drive is:

```
$> dismopt1             or     $> dismopt2
```

This will also eject the OD from the drive. Please be sure to return the OD to its proper storage place as a courtesy to other users.

After you have copied all your MRI data to the appropriate working directory, you are ready to begin working with this information!

A note about optical disks used at the BIC:

MRI data is stored on low density (512 bytes per sector) optical disks and used only in the optical disk drive on the SGI computer known as Bottom. PET data is stored on high density (1024 bytes per sector) optical disks and used *only* in the optical disk drive on the Sun computer attached to the PET scanner (known as the ECAT HR+). Have someone like the BIC system administrator (presently Jean-Francois Malouin, Sylvain Milor or Dale Einarson as of Feb. 4, 2004) or one of the PET analysis staff (Sylvain Milot) show you where the optical disks and OD drives for each of these computers are located before you start loading things up.

AUTOMATED IMAGE PROCESSING AND QUALITY CONTROL

Now, you must register the PET and MRI information. Image registration does two things:

a) corrects for subject motion from one PET scan to another and for
   motion during a PET or MRI scan and

b) correlates each subject's PET and MRI data

it is this second step that allows one to look at PET information overlaid on the MRI volume, thereby making the PET activation data more visually meaningful. Please see the article by Collins et al. (1994) JCAT 18(2) 192-205 for a complete discussion of the image registration processes used at the BIC.

From the "base" directory (in this case, we will use /data/aging/nling/mystudy5_dat):

```
$> do_mritotal -resample all -verify all -batch -noexec
```

The -noexec option tells the program not to execute the commands, just to tell you what it will do when it actually carries out the command and options you specified. This allows you to see whether the system recognizes the MRI files as correctly acquired and named; it also shows where the results will be stored (normally in the directories mrital_xfm and mrital_rsl; the transform files will be in the form lastname_firstname_mrital.xfm and will be located in the mrital_xfm directory while the resampled mri files will be of the form lastname_firstname_mrital.mnc and will be found in the directory mrital_rsl). NB: if do_mritotal does not complete for every subject, check the log files to see why the process didn't finish. An easy work-around is, if you have *.gz files, to uncompress them using

```
$> gunzip *.gz.
```

Here is an example of the command do_mritotal with the -noexec option specified:

```
bottom:/data/aging/nling/mystudy5_dat>do_mritotal -noexec -resample all -verify
all -batch
Checking for properly acquired T1 MRI data...
  mri/bento_phil_9139000_2_0120_mri.mnc.gz
  mri/bubble_aurora_9396000_2_0120_mri.mnc.gz
  mri/smith_mary_8954000_2_0120_mri.mnc.gz
  mri/jones_anthony_9030000_2_0120_mri.mnc.gz
  mri/hierophant_sylvana_9313000_2_0120_mri.mnc.gz
```



```
  mri/jackson_marshall_9404000_2_0120_mri.mnc.gz
  mri/mcgregor_alex_9419000_2_0120_mri.mnc.gz
  mri/poulin_jean_jacques_9848000_2_0120_mri.mnc.gz
  mri/olivetti_thomas_9384000_2_0120_mri.mnc.gz
  mri/singer_elvira_9171000_2_0120_mri.mnc.gz
  mri/ricoh_richard_8827000_2_0120_mri.mnc.gz
  mri/norton_frederick_9330000_2_0120_mri.mnc.gz
Nothing to do for subject bento_phil
Nothing to do for subject bubble_aurora
Nothing to do for subject smith_mary
Nothing to do for subject jones_anthony
Nothing to do for subject hierophant_sylvana
Nothing to do for subject jackson_marshall
Nothing to do for subject mcgregor_alex
Nothing to do for subject poulin_jean_jacques
Nothing to do for subject olivetti_thomas
Nothing to do for subject singer_elvira
Nothing to do for subject ricoh_richard
Nothing to do for subject norton_frederick
Redirecting output to logs/mystudy5_dat_mrital.log and detaching to background
bottom:/data/aging/nling/mystudy5_dat>
```

In this example, there are 12 subjects and it found the original MRI files in the mri directory, however, the transform files have already been generated and the MRIs have been resampled to MNI space (based on the coordinate system developed by Talairach and Tournoux (1988) Springer-Verlag publishers, Stuttgart) so there was nothing else to do.

If, after running this command, you are satisfied that it will use the correct input information and will generate the correct output, you can execute the command for real:

```
$> do_mritotal -resample all -verify all -batch
```

This may take a few hours of computing time (especially if you have not yet run the command on your data), so you may want to work on something else while you are waiting. You do not have to remain logged on in order for the system to process a job that is running in "batch" or "background" mode.

Once the job is complete you will be notified by e-mail. In that message, you will be told where to find the "verification" images. Normally, they are stored in the verify directory under the name mystudy5_enc_1.rgb and, if necessary, mystudy5_enc_2.rgb and so on. The program do_mritotal generates one verification image for every eight subjects.

To look at these images, go to the verify directory:

```
$> cd /data/aging/nling/mystudy5_enc/verify
```

And use the command display to look at the files:

```
$> display mystudy5_enc_mrital_1.rgb &

   and/or

$> display mystudy5_enc_mrital_2.rgb &

   etc.
```

QUALITY CONTROL FOR do_mritotal:

These are the things you should be looking for when examining the verification images:

a) Is the brain tissue inside the registration box?
b) Do the slices displayed look homologous to the avg_305 brain slices
   shown at the end of the row?
c) Do the slices look like they are properly oriented?

```
d) Do the slices look like they are 'clean' (i.e., not much
   noise/artifact in the image)?

$> do_mritopet -verify -resample -batch

Warning, do_mritopet is pretty smart, but it's not smart enough to
know about maiden/married names (as far as I can tell).  If
do_mritopet gives you a bad matchup between mri files and
lowest-run-number pet files, you will have to rename the mri files to
be in accordance with your pet file naming convention.  This should
_not_ interfere with your mritotal transform files, etc. when the data
is autoloaded in Dot.  More to follow on this...

$> do_pettopet -verify -background &

For do_pettopet, you must move any previously generated *petpet*
verification images from the verify directory (to, old_verify,
perhaps) before executing this command.  Otherwise the program will
halt when it attempts to create a verification image for a subject
whose data has already been processed.

--
For additional information, you can look at the help files for these
three programs:

$> do_mritotal -help
$> do_mritopet -help
$> do_pettopet -help


MINC (Medical Image Net-Cdf)

The minc file format is used for PET, MRI and fMRI data acquired at
the MNI.  One advantage of this file format is the amount of
information you can extract from the 'header' of the file.

If you type mincinfo filename.mnc, the system will return some very
basic information about the size and dimensions of the file.  For
example, I want to know about the mri minc file for subject moran:

$> mincinfo moran_karen_8615000_2_0120_mri.mnc.gz

file: moran_karen_8615000_2_0120_mri.mnc.gz
image: signed__ short 0 to 4095
image dimensions: xspace zspace yspace
    dimension name         length         step        start
    --------------         ------         ----        -----
    xspace                    161            1     -80.4921
    zspace                    256           -1        127.5
    yspace                    256           -1      167.504

To extract very specific information from a minc file, you can use the
-attval switch.  In this example, I wanted to find out the full
name, sex, birthdate and study start time for subject moran:

$> mincinfo -attval patient:full_name -attval patient:sex -attval patient:birthd
ate -attval study:start_time moran_karen_8615000_2_0120_mri.mnc.gz

CARRUTHERS KAREN
female
1915.02.02
1998.10.27 08:27:4570

Here is an example of mincinfo for a PET file:

$> mincinfo duke_tom_199808251301.mnc

file: duke_tom_199808251301.mnc
image: signed__ short -32000 to 32000
image dimensions: zspace yspace xspace
    dimension name         length         step        start
```



```
    --------------         ------         ----        -----
    zspace                     63       -2.425      -424.97
    yspace                    128        2.025     -128.588
    xspace                    128        2.025     -128.588


Here's a "foreach" function that will gather info. from all minc
headers in a directory and write it to a file:

foreach file (*_2_*.mnc.gz)
foreach? mincinfo -attvalue patient:full_name -attvalue patient:sex -attvalue pa
tient:birthdate -attvalue study:start_time -error unknown $file >> mri_info.txt
foreach? end

# Now, you just have to figure out how to insert the "no new line" switch!
```