The problem of dependences for mrtrix in CentOS 5.2 can be solved using schroot with a debian distribution that has the possibility to use the neurodebian repositories. The approximation that resulted best for us, was to compile all the programs that are needed in a virtual machine with CentOS 5.2, specifically we use virtualbox to virtualize the operative system.
First we compile schroot, so we need to download the schroot-1.1.5-4.src.rpm source package from http://www.nikhef.nl/~dennisvd/schroot.html.
In CentOS is necessary to install this libraries:

```
yum install rpmbuild  boost boost-devel pam-devel e2fsprogs-devel
lock-devel gcc-c++ make lockdev-devel gcc kernel-devel
```

```
yum groupinstall "Development Tools"
```

Then we need to install the schroot package

```
rpm -i schroot-1.1.5-4.src.rpm
```

At this point there was a problem with the installation, which we solved it with the creation of the `redhat` directory in `/usr/src/` but google found yet another solution which was not tested by us. The solution mentioned the use of the option nomd5

```
rpm --nomd5 -i schroot-1.1.5-4.src.rpm
```

After the source is install is necessary change directory to SPECS inside the redhat directory and compile schroot:

```
cd /usr/src/redhat/SPECS
rpmbuild -bb schroot.spec
```

Then in the build directory we can find the schroot rpm package and the last step is to install it

```
rpm -Uvh schroot-1.1.5-4.x86_64.rpm
```

The program debootstrap was downloaded from
http://pkgs.org/centos-5-rhel-5/epel-i386/debootstrap-1.0.7-3.el5.noarch.rpm.html
and the installation was straightforward

```
rpm -Uvh debootstrap-1.0.7-3.el5.noarch.rpm
```

After this step we create a directory for the chroot and use debootstrap to download the chroot. Is important to notice that the *lenny* distribution worked perfectly by using the archive repositories of debian. However, in order to install squeeze and have a newer version of mrtrix,

it is necessary to create a symbolic link for squeeze in the */usr/lib/debootstrap/scripts*

```
ln -s /usr/lib/debootstrap/scripts/sid
/usr/lib/debootstrap/scripts/squeeze
```

Using debootstrap to install chroot we download the entire operating system. This command installs the OS in the directory that we recently create.

For Lenny:

```
debotstrap --arch amd64 lenny lenny/ http://archive.debian.org/debian
```

For Squeeze:

```
debotstrap --arch amd64 squeeze squeeze/
http://ftp.us.debian.org/debian
```

Now we can modify the /etc/schroot/schroot.conf in order to include the chroot recently installed.

```
[lenny]
description=Debian Lenny
type=directory
location=/home/user/lenny
priority=3
users=YOUR_USER
groups=SOME_GROUP_LIKE_users
root-groups=YOUR_ADMIN_USER
run-setup-scripts=true
run-exec-scripts=true
```

In order to have internet acces is necesary to copy /etc/resolv.conf to the etc of the chroot something like:

```
cp  /etc/resolv.conf /home/user/lenny/etc/
```

To get inside chroot we need to use the command :

```
schroot -c lenny
```

and it will give us a debian terminal where we can update and upgrade the system.

To install mrtrix we need to use the neurodebian repository

```
>wget -O- http://neuro.debian.net/lists/squeeze.de-m.full | sudo tee
/etc/apt/sources.list.d/neurodebian.sources.list
>sudo apt-key adv --recv-keys --keyserver pgp.mit.edu 2649A5A9
```

If we use Lenny is necessary to change the neurodebian.sources.list and change squeeze for lenny.

```
deb http:\\neurodebian.ovgu.de/debian lenny main contrib non free
```

Now we can install mrtrix by typing:

```
>sudo apt-get update

>sudo apt-get upgrade

>sudo apt-get install mrtrix
```

We can add the path for the binaries and libraries

```
>export
PATH=/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/
bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/lib/mrtrix/bin

> export LD_LIBRARY_PATH=/usr/lib/mrtrix/lib
```

At this point is easy to run the mrtrix binaries inside chroot, but it is possible to run them outside the chroot:

```
schroot -p -c lenny/  /usr/lib/mrtrix/bin/streamtrack

   I: [lenny chroot] Running
command:"/usr/lib/mrtrix/bin/streamtrack"

   streamtrack:expected exactly 3 arguments (0 supplied)
```

So, here we are, issuing an mrtrix command from CentOS, but the actual command gets run by debian, transparently to the user. We have tested this even on a SunGridEngine cluster, and it works like a charm.

Note that if you are using a computing cluster, then schroot must be installed in every

executing node, and schroot.conf must point to the chroot, which must be accessible to the node.

For each executing node:

Install boost libraries:
`yum install boost`

Update
bzip2-libs x86_64

Also install the following dependencies:
libicu
bzip2-libs i386

And accept the GPG key:

http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

Install lockdev
`yum install lockdev`

Install schroot
`rpm -Uvh schroot-1.1.5-4.x86_64.rpm`

and, finally, edit etc/schroot/schroot.conf to point to the chroot, as described above. This node is ready to run debian!