Summer School Checklist

Here are several things you can check to make sure you are ready for the summer school.

Attendee Checklist

- Download and install the FermiBottle container following the Installation instructions.
- Start and attach to the container following the usage instructions.
- Once in the container.
    - Check that X-windows works by, e.g., launching ds9 (or xeyes for more fun).
    - Check that you can see the shared directory in both the container and system OS.
    - Activate the 'fermi' environment with conda.
    - Check that Jupyter notebooks work.
    - Check that Fermitools work.
    - Check that the GBM Data Tools work.

To help, we've created a run through below and short Jupyter notebook you can download to demonstrate these steps.

Fermitools Example Run Through

Here's an example session on a Mac. I created the shared directory and a file in it called *this_is_the_share_directory* just so something shows up in a directory listing. I then pulled the FermiBottle image from Docker and started it.

```
(base) user@host % mkdir shared
(base) user@host % cd shared
(base) user@host % touch this_is_the_share_directory
(base) user@host % ls
this_is_the_share_directory
(base) user@host % docker pull fssc/fermibottle
Pulling from fssc/fermibottle
Digest: sha256:10c84c7355b26fdb2dba3ff312028033ac7089868715992cbf0c77fba3bc369e
 Status: Image is up to date for fssc/fermibottle
docker.io/fssc/fermibottle
(base) user@host fermi_summer_school % xhost + 127.0.0.1 && \
docker create -it --init \
-e HOST_USER_ID=`id -u $USER` \
-e DISPLAY=host.docker.internal:0 \
-v "$HOME/shared":/shared \
-p 8888:8888 \
fssc/fermibottle
127.0.0.1 being added to access control list
be68136958ad303810707e78b08080c20490fcdc2ee4b061fb46f464b997e7a4
(base) user@host % docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS    PORTS     NAMES
be68136958ad   fssc/fermibottle   "/opt/docker/bin/ent…"   19 seconds ago   Created             silly_pasteur
(base) user@host % docker start be68136958ad
be68136958ad
(base) user@host % docker attach be68136958ad
```

You can use either the container id or name of the container to start and attach to it.

This puts me in the container. You should notice that the prompt is now the "fermi" user and the container ID. Note that you only have to run the xhost command once. If you quit the container, you just have to run the start and attach commands to return to it.

I launched ds9 to make sure X-windows are working. Quit it (assuming it works) and change to the shared directory, which is under */shared* in the container.

```
(base) [fermi@be68136958ad ~]$ ds9
(base) [fermi@be68136958ad shared]$ cd /shared
(base) [fermi@be68136958ad shared]$ ls
this_is_the_share_directory
```

Back on the **host system**, download this file, Quick_Test.zip, and put it in the shared directory. Then, unzip it which will create a directory.

```
(base) user@host shared % unzip Quick_Test.zip
Archive:  Quick_Test.zip
   creating: Quick_Test/
  inflating: Quick_Test/Quick_Test.ipynb
  inflating: Quick_Test/SC00.fits
  inflating: Quick_Test/3C279_1deg_gti.fits
(base) user@host fermi_summer_school % ls
Quick_Test/                 Quick_Test.zip              this_is_the_share_directory
```

In the **container**, you should see the change:

```
(base) [fermi@be68136958ad shared]$ ls
Quick_Test  Quick_Test.zip  this_is_the_share_directory
```

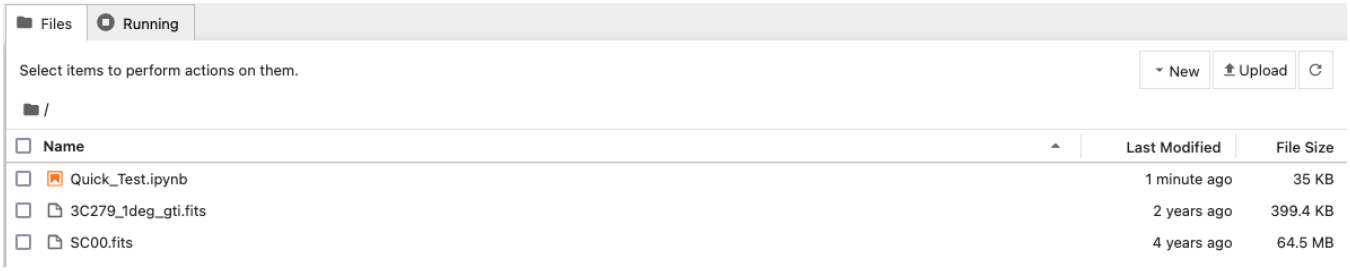
Now start the "fermi" environment and the Jupyter notebook server:

```
(base) [fermi@be68136958ad shared]$ conda activate fermi
(fermi) [fermi@be68136958ad shared]$ cd Quick_Test
(fermi) [fermi@be68136958ad Quick_Test]$ ls
3C279_1deg_gti.fits  3C279_1deg_lc.fits  Quick_Test.ipynb  SC00.fits

(fermi) [fermi@be68136958ad Quick_Test]$ notebook
    <various output not shown>
    To access the server, open this file in a browser:
        file:///home/fermi/.local/share/jupyter/runtime/jpserver-328-open.html
    Or copy and paste one of these URLs:

        http://127.0.0.1:8888/tree?token=2d13b3d5a3c4e69177742f8df5ebeaba475c8a5703042327
```

You can paste one of the http URLs into the browser (I used the 00.00.00.00 one) on your host machine and should see something like:



Click on the *Quick_test.ipynb* file to open it. You can then run the notebook by select **RunRun All Cells** from the menu bar. It runs one of the Fermitools (gtbin) and then creates a lightcurve. You should then end up with a notebook that looks like this:

Quick Fermi Container Test

This notebook tests the Fermitools functionality by using gtbin to make a lightcurve. It should not take very long to run.

[1]:
```
from astropy.io import fits as pyfits
%pylab inline
import gt_apps
from gt_apps import evtbin
```

```
%pylab is deprecated, use %matplotlib inline and import the required libraries.
Populating the interactive namespace from numpy and matplotlib
```

[2]:
```
evtbin['algorithm'] = 'LC'
evtbin['outfile'] = '3C279_1deg_lc.fits'
evtbin['evfile'] = '3C279_1deg_gti.fits'
evtbin['scfile'] = 'SC00.fits'
evtbin['tbinalg'] = 'LIN'
evtbin['tstart'] = 239557417
evtbin['tstop'] = 255398400
evtbin['dtime'] = 86400
```

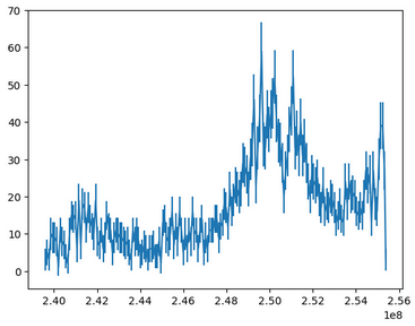
[3]:
```
evtbin.run()
```

```
time -p gtbin evfile=3C279_1deg_gti.fits scfile=SC00.fits outfile=3C279_1deg_lc.fits algorithm="LC" ebinalg="LOG" emin=30.0 emax=200000.0 enu
mbins=0 denergy=0.0 ebinfile=NONE tbinalg="LIN" tstart=239557417.0 tstop=255398400.0 dtime=86400.0 tbinfile=NONE snratio=0.0 lcemin=0.0 lcema
x=0.0 nxpix=0 nypix=0 binsz=0.0 coordsys="CEL" xref=0.0 yref=0.0 axisrot=0.0 rafield="RA" decfield="DEC" proj="AIT" hpx_ordering_scheme="RING
" hpx_order=3 hpx_ebin=yes hpx_region="" evtable="EVENTS" sctable="SC_DATA" efield="ENERGY" tfield="TIME" chatter=2 clobber=yes debug=no gui=
no mode="ql"
This is gtbin version HEAD
real 0.36
user 0.24
sys 0.05
```

[4]:
```
lc_gtbin = pyfits.open('3C279_1deg_lc.fits')
```

[5]:
```
errorbar(lc_gtbin[1].data.field('TIME'),lc_gtbin[1].data.field('Counts'),yerr=lc_gtbin[1].data.field('Error'))
```

[5]:
```
<ErrorbarContainer object of 3 artists>
```



Congratulations, things seem to be working!

GBM Data Tools Example

The GBM Data Tools tutorials are in Jupyter notebooks. If you did the example above, you already have a notebook running, but you can't have more than one Jupyter notebook running at a time in FermiBottle by default. You can use the "Quit" button in the upper right of the Jupyter notebook start page or hit to `ctrl-c` in the terminal window, which will then prompt you to shut it down.

The GBM Data Tools are in separate conda environment from the Fermitools. The GBM team provides the *gbm-demos* command to launch a list of the tutorial notebooks.

```
(fermi) [fermi@be68136958ad Quick_Test]$ conda deactivate
(base) [fermi@be68136958ad Quick_Test]$ conda activate fermigbm
(fermigbm) [fermi@be68136958ad Quick_Test]$ gbm-demos
```

Like before, you will need to paste the URL into an external browser and will then find a list of notebooks. Open the *index.ipynb* one, which gives more descriptive list of the available tutorials. Try the first one on the list "GBM Science Data: Time History Spectra" and do **RunRun All Cells**. It should run quickly and make some plots. You may see a warning that Basemap is not installed. That shouldn't be a problem for now.