Load needed libraries.

In [2]:
```
import numpy
import astropy
import sunpy
import sunpy.map
from scidbpy import connect, SciDBQueryError, SciDBArray
from matplotlib import pyplot as plt
```

Connect to SciDB

In [3]:
```
sdb = connect('http://localhost:8080')
afl = sdb.afl
```

Load a 3D array with AIA pictures of a specific Flare. Print schema.

In [5]:
```
sdo= sdb.wrap_array("Flare_128970")
print sdo.datashape.schema
```

```
<aia94:float,aia131:float,aia171:float,aia193:float,aia211:float,aia304:fl
oat,aia335:float,hmi_magnetogram:float> [x=0:4095,256,2,y=0:4095,256,2,tim
e=0:*,1,0]
```

Select only 1 AIA131 Picture out of the time-series. Eval() due to the lazy evaluation of SciDB.

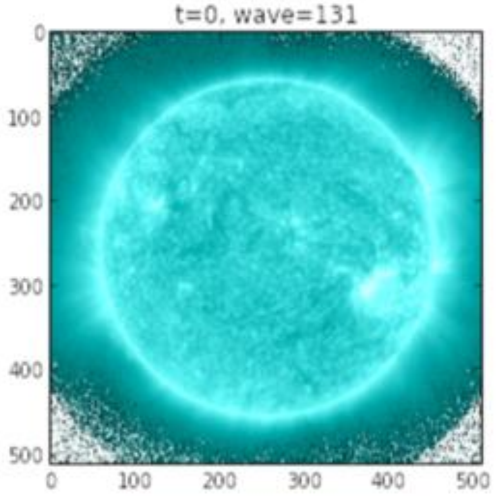
In [7]:
```
tmp131=sdo.subarray(0,0,0,4095,4095,0).project('aia131')
tmp131.eval()
print tmp131.datashape.schema
print tmp131.name
```

```
<aia131:float> [x=0:4095,256,2,y=0:4095,256,2,time=0:0,1,0]
py1100966044621_00002
```

Use matplotlib to visualize the data. Regrid() will resize the image in order to speed up the data transfer.

In [8]:
```
plt131=afl.regrid('py1100966044621_00002','8,8,1',
                  'avg(aia131) as avg_aia131').reshape((512,512))
plt.title('t=0, wave=131')
plt.imshow(plt131.toarray(), norm=matplotlib.colors.LogNorm(),
           cmap=plt.get_cmap("sdoaia131"))
plt.show
```

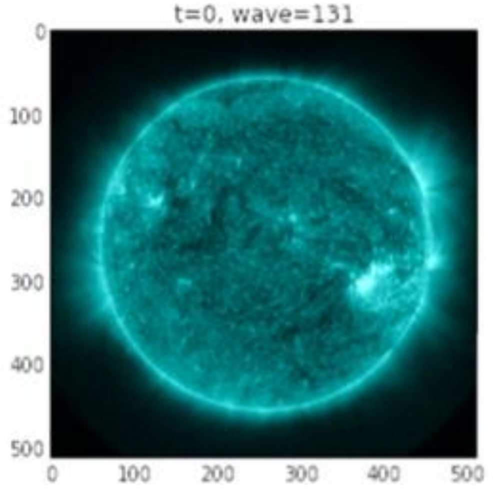
```
Out[8]: <function matplotlib.pyplot.show>
```



Do a "preprocessing" of the data and convert all values < 1 to 1 in order to remove the noise.

```
In [16]: tmp131_clean=tmp131.apply('aia131_clean','iif(aia131<1,1,aia131)')
                              .project('aia131_clean')
         tmp131_clean.eval()
         print tmp131_clean.name

         py1100966044621_00018
```

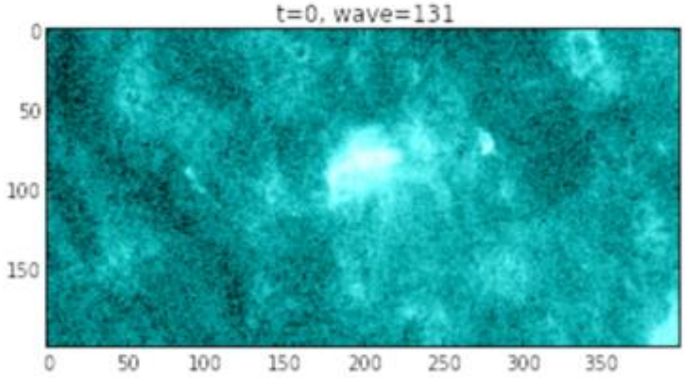
```
In [17]: plt131_clean=afl.regrid('py1100966044621_00018','8,8,1',
                                 'avg(aia131_clean) as avg_aia131')
                                 .reshape((512,512))
         plt.title('t=0, wave=131')
         plt.imshow(plt131_clean.toarray(), norm=matplotlib.colors.LogNorm(),
                    cmap=plt.get_cmap("sdoaia131"))
         plt.show

Out[17]: <function matplotlib.pyplot.show>
```



Print only a subarray of the picture useing subarray(). Use reshape to convert 3D to 2D but with no reduction this time

```
In [18]: plt131_clean=tmp131_clean.reshape((4096,4096))
         plt.title('t=0, wave=131')
         plt.imshow(plt131_clean[1650:1850,3050:3450].toarray(),
                    norm=matplotlib.colors.LogNorm(),cmap=plt.get_cmap("sdoaia131"))
         plt.show
```

```
Out[18]: <function matplotlib.pyplot.show>
```



Extract the limb of the sun useing the euclid-distance.

```
In [20]: tmp131_limb=tmp131_clean.apply('aia_limb','iif(sqrt(pow(y-2048,2)'+
                      '+pow(x-2048,2))<1808 AND sqrt(pow(y-2048,2)'+
                      '+pow(x-2048,2))>1448,aia131_clean,0)').project('aia_limb')
         tmp131_limb.eval()
         print tmp131_limb.name
```

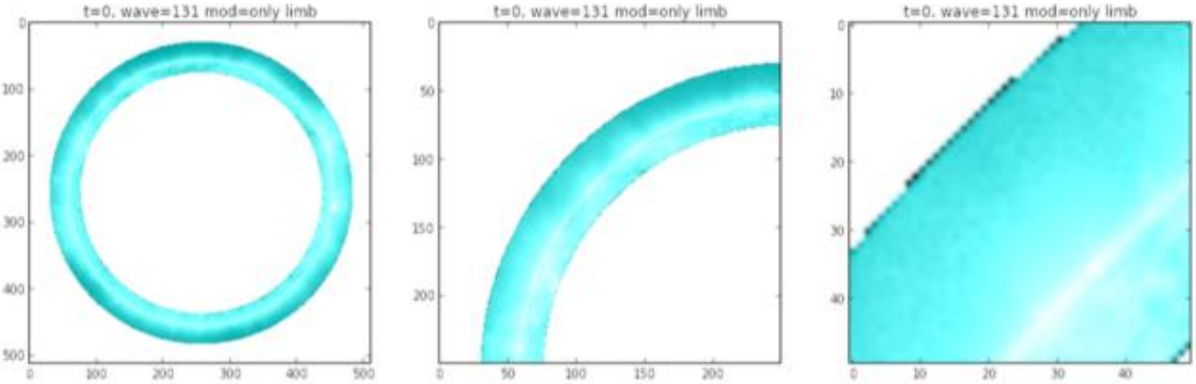
```
py1100966044621_00025
```

In [25]:

```
plt131_limb_clean=afl.regrid('py1100966044621_00025','8,8,1',
                        'avg(aia_limb) as aia_limb').reshape((512,512))
plt.figure(1,figsize=(17,6))
plt.subplot(131)
plt.title('t=0, wave=131 mod=only limb')
plt.imshow(plt131_limb_clean.toarray(),
            norm=matplotlib.colors.LogNorm(),cmap=plt.get_cmap("sdoaia131"))
plt.subplot(132)
plt.title('t=0, wave=131 mod=only limb')
plt.imshow(plt131_limb_clean[0:250,0:250].toarray(),
            norm=matplotlib.colors.LogNorm(),cmap=plt.get_cmap("sdoaia131"))
plt.subplot(133)
plt.title('t=0, wave=131 mod=only limb')
plt.imshow(plt131_limb_clean[80:130,80:130].toarray(),
            norm=matplotlib.colors.LogNorm(),cmap=plt.get_cmap("sdoaia131"))
plt.show
```

Out[25]:

```
<function matplotlib.pyplot.show>
```



Create an aggregate for each picture of a subarray over time. Use aggregate to to do the calculations.

In [26]:

```
tmp_flare=sdo.subarray(1650,3250,0,1850,3450,16)
tmp_flare.eval()
```

Out[26]:

```
SciDBArray('py1100966044621_00052<aia94:float,aia131:float,aia171:float,ai
a193:float,aia211:float,aia304:float,aia335:float,hmi_magnetogram:float>
[x=0:200,256,2,y=0:200,256,2,time=0:16,1,0]')
```

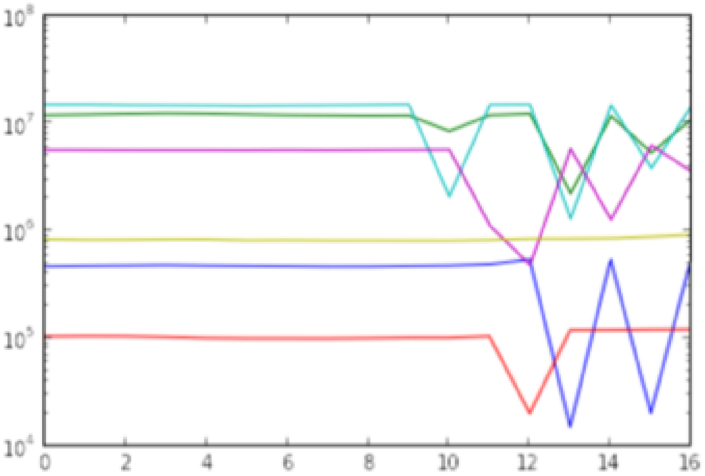
In [27]:

```
flare_aggregated=afl.aggregate('py1100966044621_00052',
                    'sum(aia131)','sum(aia171)','sum(aia94)','sum(aia193)',
                    'sum(aia211)','sum(aia304)','time')
flare_aggregated.eval()
print flare_aggregated.datashape.schema
```

```
<aia131_sum:double NULL DEFAULT null,aia171_sum:double NULL DEFAULT null,a
ia94_sum:double NULL DEFAULT null,aia193_sum:double NULL DEFAULT null,aia2
11_sum:double NULL DEFAULT null,aia304_sum:double NULL DEFAULT null> [tim
e=0:16,1,0]
```

In [28]:

```
plt.plot(flare_aggregated['aia131_sum'].toarray(),label="aia131")
plt.plot(flare_aggregated['aia171_sum'].toarray(),label="aia171")
plt.plot(flare_aggregated['aia94_sum'].toarray(),label="aia94")
plt.plot(flare_aggregated['aia193_sum'].toarray(),label="aia193")
plt.plot(flare_aggregated['aia211_sum'].toarray(),label="aia211")
plt.plot(flare_aggregated['aia304_sum'].toarray(),label="aia304")
plt.yscale('log')
plt.show()
```



Create a new image based on three AIA images. Use algorithm of paper. Print some data and diffrent pictures of it.

In [34]:

```
aia018=sdo.subarray(0,0,0,4095,4095,0).reshape((4096,4096))
aia018=aia018.apply('aia018','aia94-(aia211/120)-(aia171/450)')
aia018.eval()
print aia018.name
print aia018['aia018'][2000:2003,2000:2008].toarray()
```

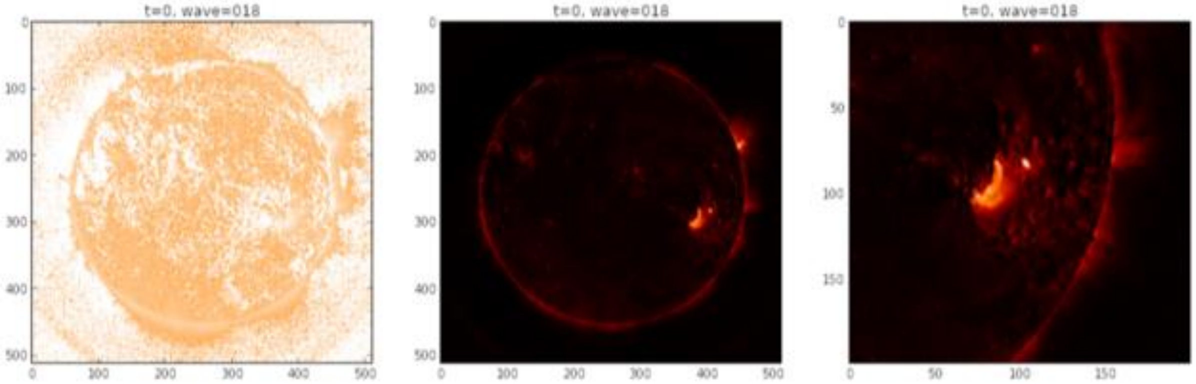
```
py1100966044621_00068
[[ 0.805       0.81611111 -0.31166667 -1.09888889 -0.18833333 -1.15111111
   0.81111111  0.78944444]
 [-1.27611111 -0.16777778  3.84222222 -2.30277778 -0.22222222 -2.20666667
  -0.24        0.74      ]
 [-0.14166667 -2.23166667  0.95222222 -1.12388889 -1.33611111 -0.22444444
  -0.20166667  0.82277778]]
```

In [31]:

```
aia018_clean=aia018.apply('aia018_clean',
                          'iif(aia018<1,1,aia018)').project('aia018_clean')
aia018_clean.eval()
print aia018_clean.name
```

```
py1100966044621_00075
```

```
In [37]: plt018=afl.regrid('py1100966044621_00068','8,8',
                           'avg(aia018) as avg_aia018')
         plt018_clean=afl.regrid('py1100966044621_00075','8,8',
                                 'avg(aia018_clean) as avg_aia018_clean')
         plt.figure(1,figsize=(17,6))
         plt.subplot(131)
         plt.title('t=0, wave=018')
         plt.imshow(plt018.toarray(), norm=matplotlib.colors.LogNorm(),
                    cmap=plt.get_cmap("sdoaia304"))
         plt.subplot(132)
         plt.title('t=0, wave=018')
         plt.imshow(plt018_clean.toarray(), norm=matplotlib.colors.LogNorm(),
                    cmap=plt.get_cmap("sdoaia304"))
         plt.subplot(133)
         plt.title('t=0, wave=018')
         plt.imshow(plt018_clean[200:400,300:500].toarray(),
                    norm=matplotlib.colors.LogNorm(),cmap=plt.get_cmap("sdoaia304"))
         plt.show

Out[37]: <function matplotlib.pyplot.show>
```



Disconnect from SciDB.

```
In [38]: sdb.reap()
```