Import all needed libraries

In [1]:
```
import numpy
import astropy
import time
from scidbpy import connect, SciDBQueryError, SciDBArray
from matplotlib import pyplot as plt
```

Conenct to SciDB Database

In [2]:
```
sdb = connect('http://localhost:8080')
afl = sdb.afl
```

Load two arrays (One HMI Picture). Show the difference. One uses overlaps.

In [3]:
```
hmi = sdb.wrap_array("HMI_test")
hmi_overlap = sdb.wrap_array("HMI_solo_overlap")
print "Size:", hmi.size
print "Shape:", hmi.shape
print "Schema:",hmi.datashape.schema
print "Schema:",hmi_overlap.datashape.schema
```

```
Size: 16777216
Shape: (4096, 4096)
Schema: <val:double> [x=0:4095,4096,0,y=0:4095,4096,0]
Schema: <val:float> [x=0:4095,128,2,y=0:4095,128,2]
```

Get some information about the values of the array.

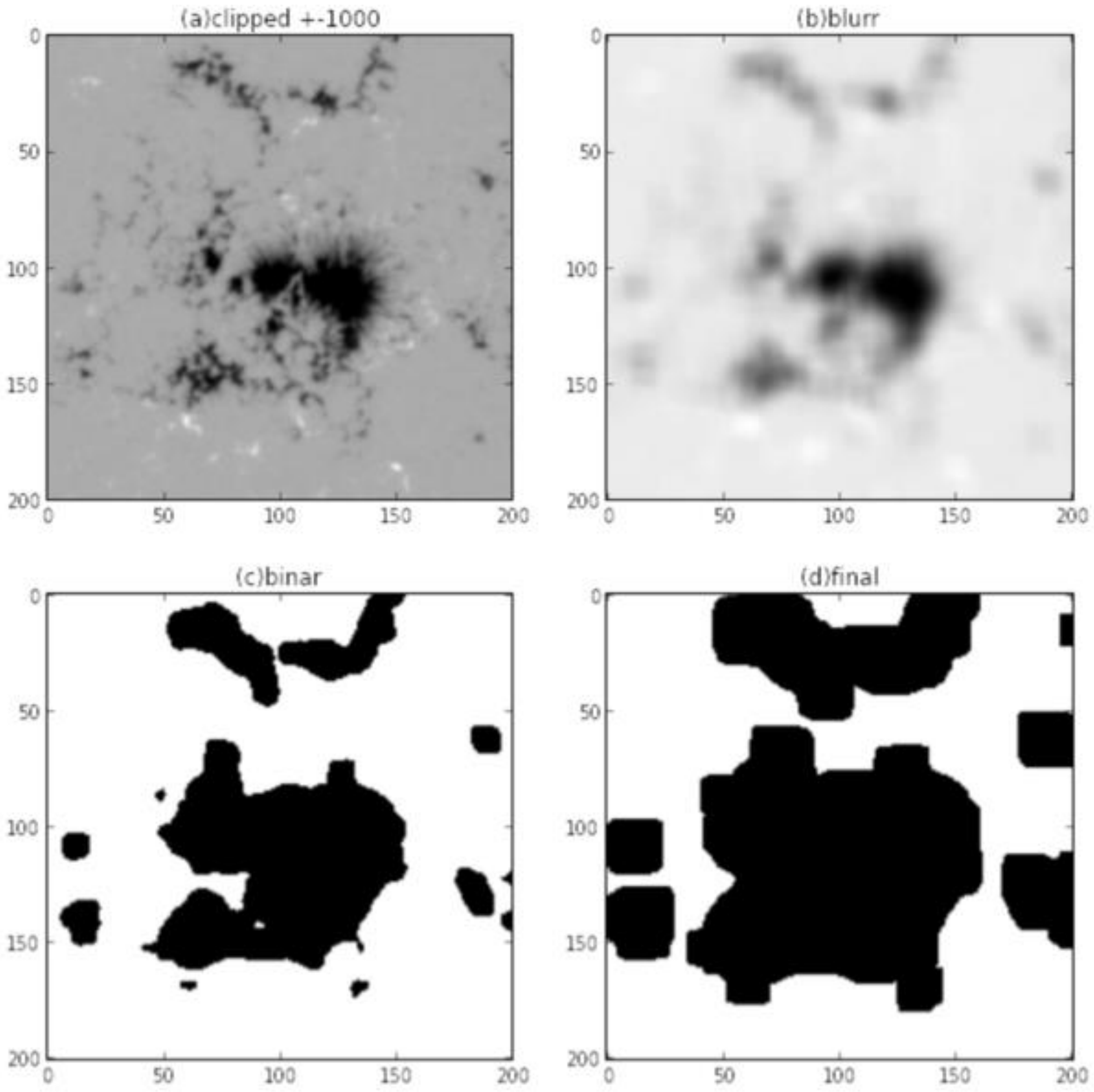
In [4]:
```
print "Max on disk:", hmi_overlap[1500:2500,1500:2500].max().toarray()
print "Min on disk:", hmi_overlap[1500:2500,1500:2500].min().toarray()
```

```
Max on disk: [ 2077.19995117]
Min on disk: [-1726.59997559]
```

Implement a part of the SMAR algorithm. A) Clip data to +- 1000 Gauss. B) Simple gaussian-blur. C) Convert to binary values. D) Grow function

In [11]:
```
hmi_a=hmi_overlap.subarray(2250,2250,2450,2450).apply('clipped',
       'iif(val<-1000,-1000,iif(val>1000,1000,val))').project('clipped')
hmi_b=hmi_a.window(5,5,5,5,'avg(clipped) as clipped').eval()
hmi_c=hmi_b.apply('binar','iif(clipped<-70,1,iif(clipped>70,1,0))')
    .project('binar').eval()
hmi_d=hmi_c.window(6,6,6,6,'max(binar) as grow').eval()
```

In [12]:
```
plt.figure(1,figsize=(10,10))
plt.subplot(221)
plt.title('(a)clipped +-1000')
plt.imshow(hmi_a.toarray(), cmap=plt.get_cmap('gray'))
plt.subplot(222)
plt.title('(b)blurr')
plt.imshow(hmi_b.toarray(), cmap=plt.get_cmap('gray'))
plt.subplot(223)
plt.title('(c)binar')
plt.imshow(hmi_c.toarray(), cmap=plt.get_cmap('binary'))
plt.subplot(224)
plt.title('(d)final')
plt.imshow(hmi_d.toarray(), cmap=plt.get_cmap('binary'))
plt.show()
```



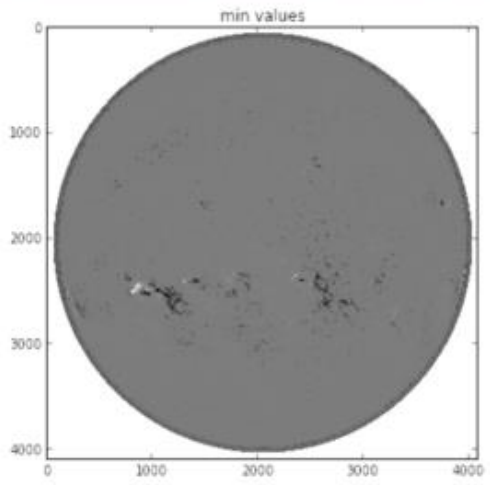
Load a time series of pictures and aggregate the min and max values over it. Print the max and min image.

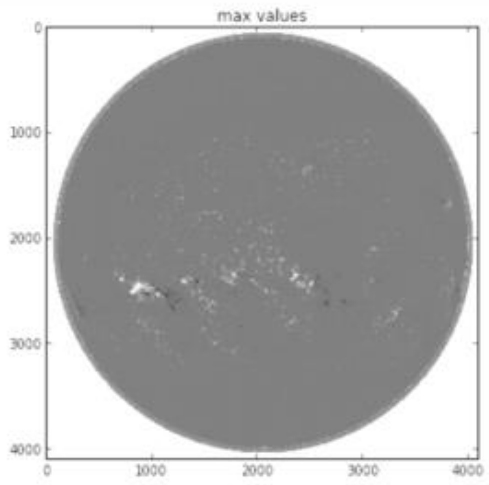
```
In [13]: hmi_cube= sdb.wrap_array("HMI_Cube")
         hmi_aggregated=hmi_cube.aggregate('min(val) as minVal',
                                           'max(val) as maxVal','x','y')
         hmi_aggregated.eval()
         print hmi_cube.datashape.schema
         print hmi_aggregated.datashape.schema
```

```
<val:float> [x=0:4095,512,1,y=0:4095,512,1,time=0:*,1,0]
<minVal:float NULL DEFAULT null,maxVal:float NULL DEFAULT null> [x=0:409
5,512,0,y=0:4095,512,0]
```

```
In [14]: plt.figure(3,figsize=(17,6))
         plt.subplot(121)
         plt.title('min values')
         plt.imshow(hmi_aggregated['minVal'].toarray(), cmap=plt.get_cmap('gray'),
                    vmin=-800, vmax=+800)
         plt.subplot(122)
         plt.title('max values')
         plt.imshow(hmi_aggregated['maxVal'].toarray(), cmap=plt.get_cmap('gray'),
                    vmin=-800, vmax=+800)
         plt.show()
```

```
/usr/lib/pymodules/python2.7/matplotlib/colors.py:533: RuntimeWarning: inv
alid value encountered in less
  cbook._putmask(xa, xa<0.0, -1)
```





Disconnect from SciDB.

```
In [15]: sdb.reap()
```