Load all needed libraries

In [1]:
```
import numpy
from scidbpy import connect, SciDBQueryError, SciDBArray
import matplotlib.pyplot as plt
%matplotlib inline
```

Conenct to database

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

Load GOES time-array (1D)

In [3]:
```
goes1 = sdb.wrap_array("GOES_1D")
print goes1.datashape.schema
```

```
<low:float,high:float> [time=0:*,250000,0]
```

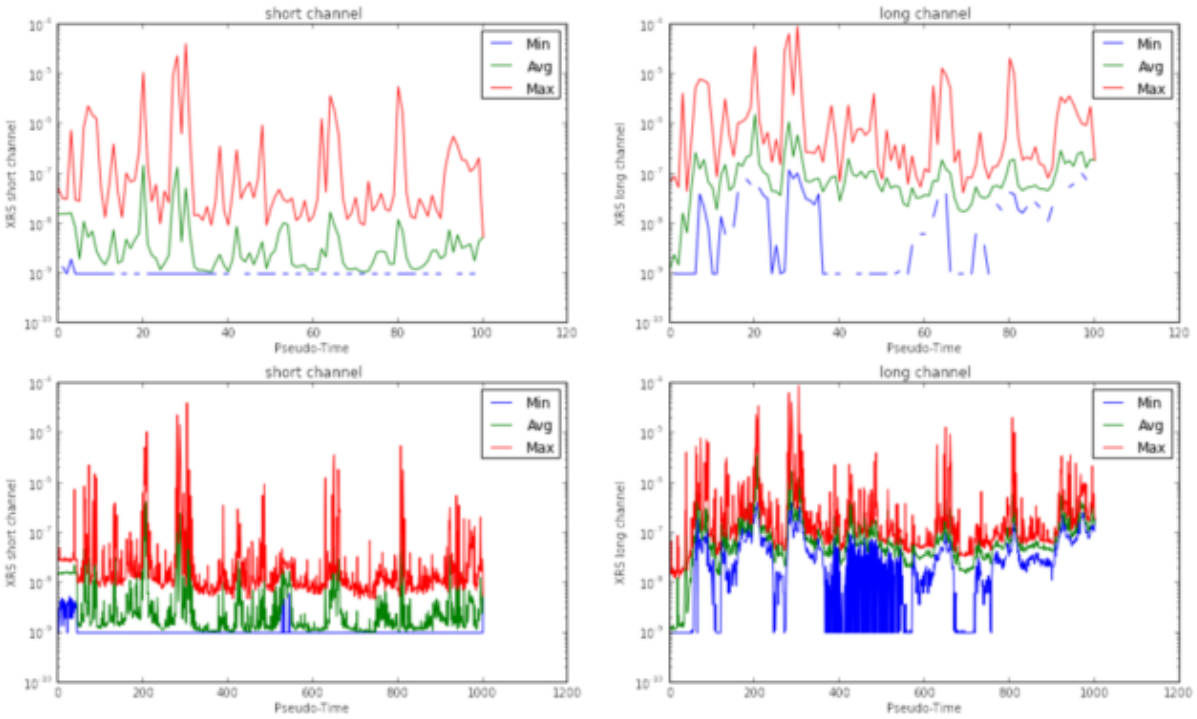
Calculate min.max,avg of blocks of 100'000 and 10'000 for both channels with the regrid oeprator.

In [4]:
```
regrid_array=afl.regrid('GOES_1D',100000,'min(low) as MinLow',
             'max(low) as MaxLow','avg(low) as AvgLow',
             'min(high) as MinHigh','max(high) as MaxHigh',
             'avg(high) as AvgHigh')
regrid_array.eval()
regrid_array_small=afl.regrid('GOES_1D',10000,'min(low) as MinLow',
                   'max(low) as MaxLow','avg(low) as AvgLow',
                   'min(high) as MinHigh','max(high) as MaxHigh',
                   'avg(high) as AvgHigh')
regrid_array_small.eval()
```

Out[4]:
```
SciDBArray('py1101455645108_00002<MinLow:float NULL DEFAULT null,MaxLow:fl
oat NULL DEFAULT null,AvgLow:double NULL DEFAULT null,MinHigh:float NULL D
EFAULT null,MaxHigh:float NULL DEFAULT null,AvgHigh:double NULL DEFAULT nu
ll> [time=0:1024,250000,0]')
```

Print the accumulated data in two plots for each aggregate/regrid.

In [5]:
```
plt.figure(1,figsize=(17, 10))
plt.subplot(221)
plt.title('short channel')
plt.plot(regrid_array['MinLow'].toarray(),label="Min")
plt.plot(regrid_array['AvgLow'].toarray(),label="Avg")
plt.plot(regrid_array['MaxLow'].toarray(),label="Max")
plt.legend(['Min','Avg', 'Max'], loc='upper right')
plt.yscale('log')
plt.ylabel('XRS short channel')
plt.xlabel('Pseudo-Time')
plt.subplot(222)
plt.title('long channel')
plt.plot(regrid_array['MinHigh'].toarray(),label="Min")
plt.plot(regrid_array['AvgHigh'].toarray(),label="Avg")
plt.plot(regrid_array['MaxHigh'].toarray(),label="Max")
plt.legend(['Min','Avg', 'Max'], loc='upper right')
plt.yscale('log')
plt.ylabel('XRS long channel')
plt.xlabel('Pseudo-Time')
plt.subplot(223)
plt.title('short channel')
plt.plot(regrid_array_small['MinLow'].toarray(),label="Min")
plt.plot(regrid_array_small['AvgLow'].toarray(),label="Avg")
plt.plot(regrid_array_small['MaxLow'].toarray(),label="Max")
plt.legend(['Min','Avg', 'Max'], loc='upper right')
plt.yscale('log')
plt.ylabel('XRS short channel')
plt.xlabel('Pseudo-Time')
plt.subplot(224)
plt.title('long channel')
plt.plot(regrid_array_small['MinHigh'].toarray(),label="Min")
plt.plot(regrid_array_small['AvgHigh'].toarray(),label="Avg")
plt.plot(regrid_array_small['MaxHigh'].toarray(),label="Max")
plt.legend(['Min','Avg', 'Max'], loc='upper right')
plt.yscale('log')
plt.ylabel('XRS long channel')
plt.xlabel('Pseudo-Time')
plt.show()
```



Show the data in a table with .todataframe() or .toarray() (everything) or with .head() for a quicklook at the first 5 elements

In [6]:
```
regrid_array.head()
```

Out[6]:

	MinLow	MaxLow	AvgLow	MinHigh	MaxHigh	AvgHigh
time						
0	0.000000e+00	5.053200e-08	1.556387e-08	0.000000e+00	6.912800e-08	1.254149e-09
1	1.373600e-09	3.260400e-08	1.596485e-08	1.000000e-09	8.965700e-08	2.422489e-09
2	1.000000e-09	3.144700e-08	1.595151e-08	1.000000e-09	5.373000e-08	1.507428e-09
3	1.951900e-09	7.540800e-07	1.657825e-08	1.000000e-09	4.149400e-06	1.634411e-08
4	1.000000e-09	3.029000e-08	8.453168e-09	1.000000e-09	4.474800e-08	6.630697e-09

Disconnect from SciDB

In [7]:
```
sdb.reap()
```