2015-11-22_intro-correlation-and-modeling

November 22, 2015

In [1]:
```
from __future__ import division
from functions import *
from utils import *

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

In [2]:
```
%%javascript
IPython.OutputArea.auto_scroll_threshold = 9999;
```

```
<IPython.core.display.Javascript object>
```

In [3]:
```
mouseA = '/Volumes/DATA/DATA/Equalized Separation/2014 Oct 27/'
mouseB = '/Volumes/DATA/DATA/Equalized Separation/2014 Oct 22/'

# mouseA = '/Users/guillaume/Projects/GEVI-DATA/2014 Oct 27/'
# mouseB = '/Users/guillaume/Projects/GEVI-DATA/2014 Oct 22/'
```

In [4]:
```
dataA = Parallel(n_jobs=8)(delayed(getExpData)(i, mouseA) for i in [3,4,5,6])
dataB = Parallel(n_jobs=8)(delayed(getExpData)(i, mouseB) for i in [2,3,4,5])
```

0.1 Discarded data (from Appendix M. Baer Msc Thesis)

0.1.1 Mouse A

exp3

- r1
- r2
- r3
- r4
- r5

exp4

- r1
- r2

exp6

- r5

0.1.2 Mouse B

exp3

- r2
- r4
- r5

exp4

- r1
- r2

```
In [5]: discard = {
            'MouseA':
            {
                3 : [1,2,3,4,5],
                4 : [1,2],
                6 : [5]
            },
            'MouseB' :
            {
                3 : [2,4,5],
                4 : [1,2]
            }
        }
```

```
In [6]: goodDataA = keepGoodData(dataA, discard['MouseA'],3)
        goodDataB = keepGoodData(dataB, discard['MouseB'],2)
```

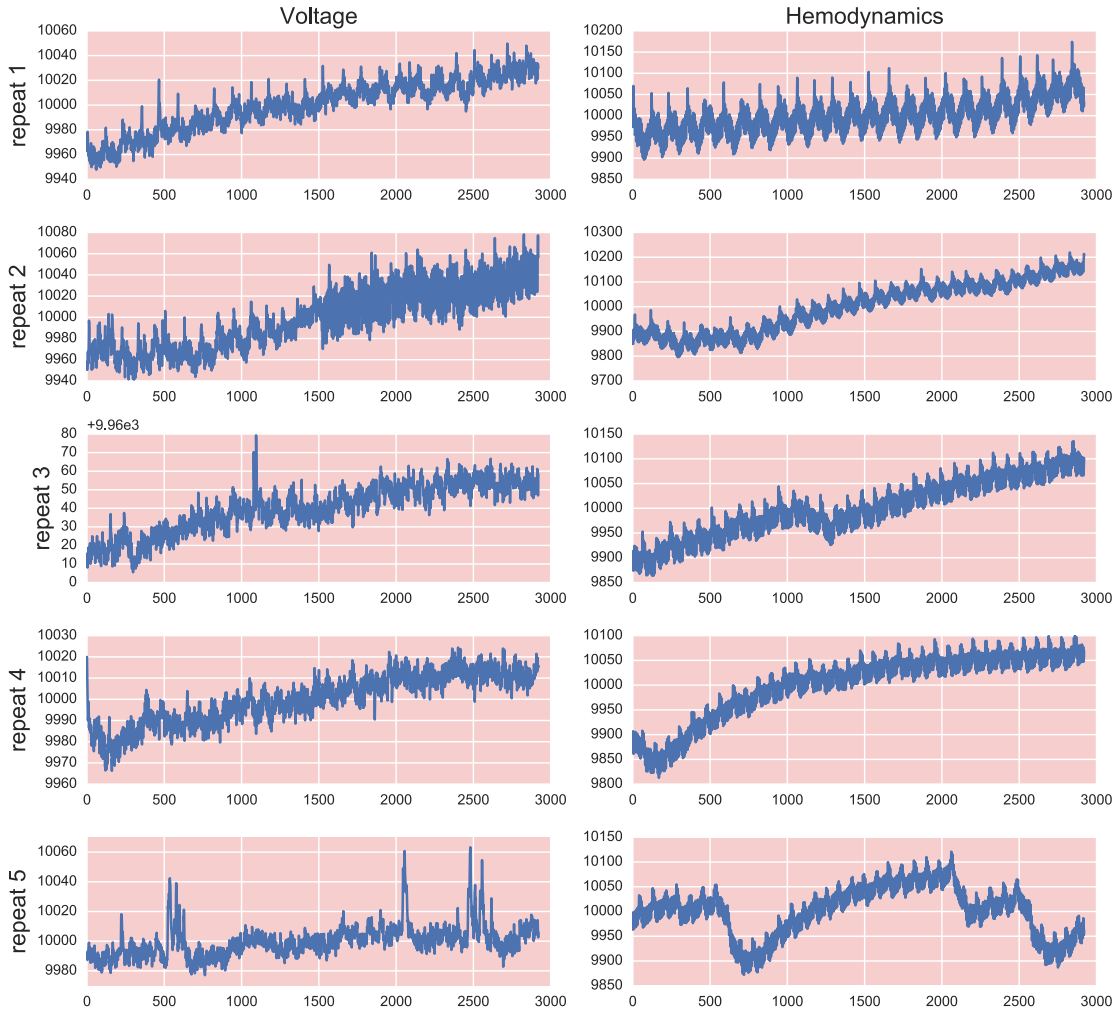
```
In [7]: # data[exp][0:ratio 1:hemo][repeat][x][y]
        np.array(goodDataA[2][0]).shape
```

```
Out[7]: (4, 2921, 60, 80)
```

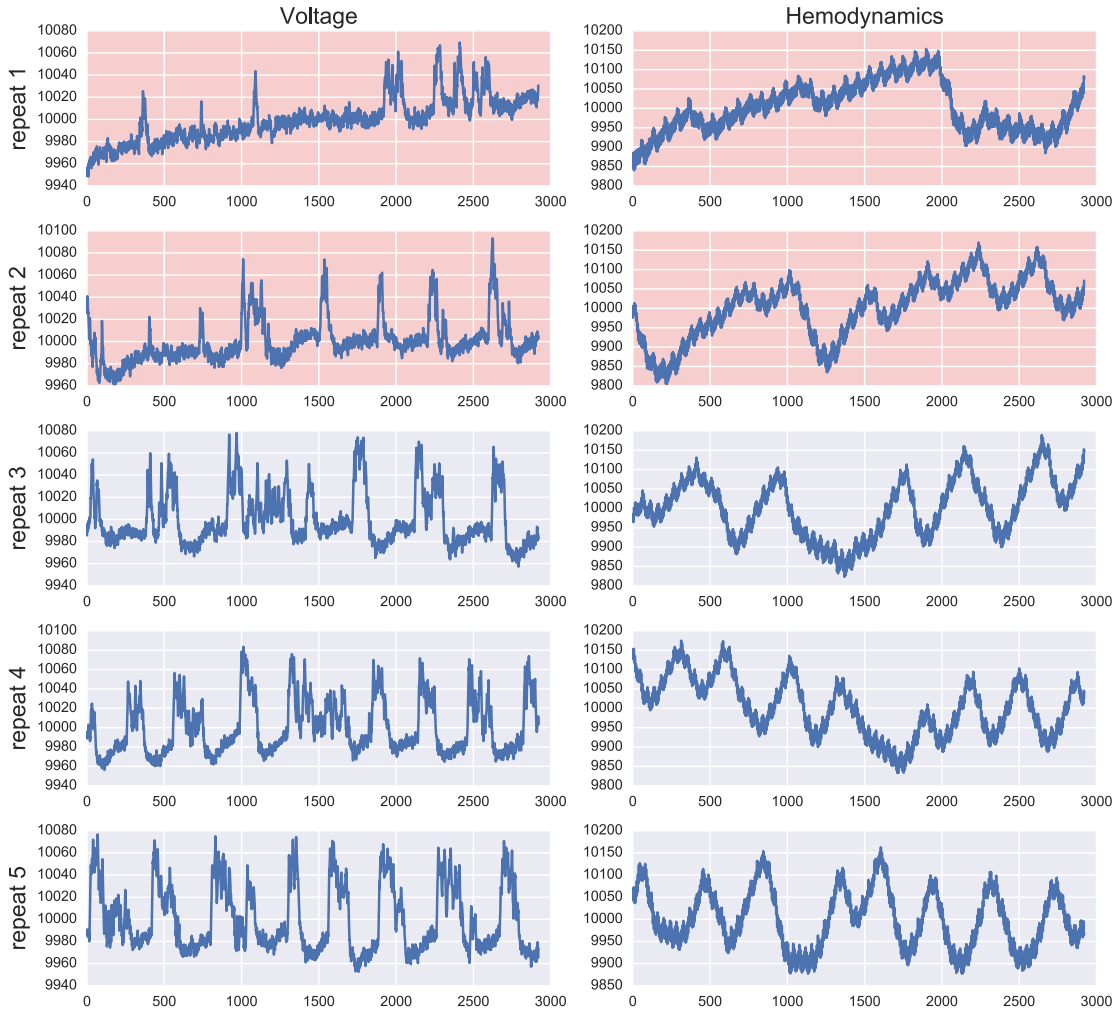
1 Mouse A

```
In [8]: for i in range(len(dataA)):
            plotData(dataA[i], i, discard['MouseA'], start=3)
```

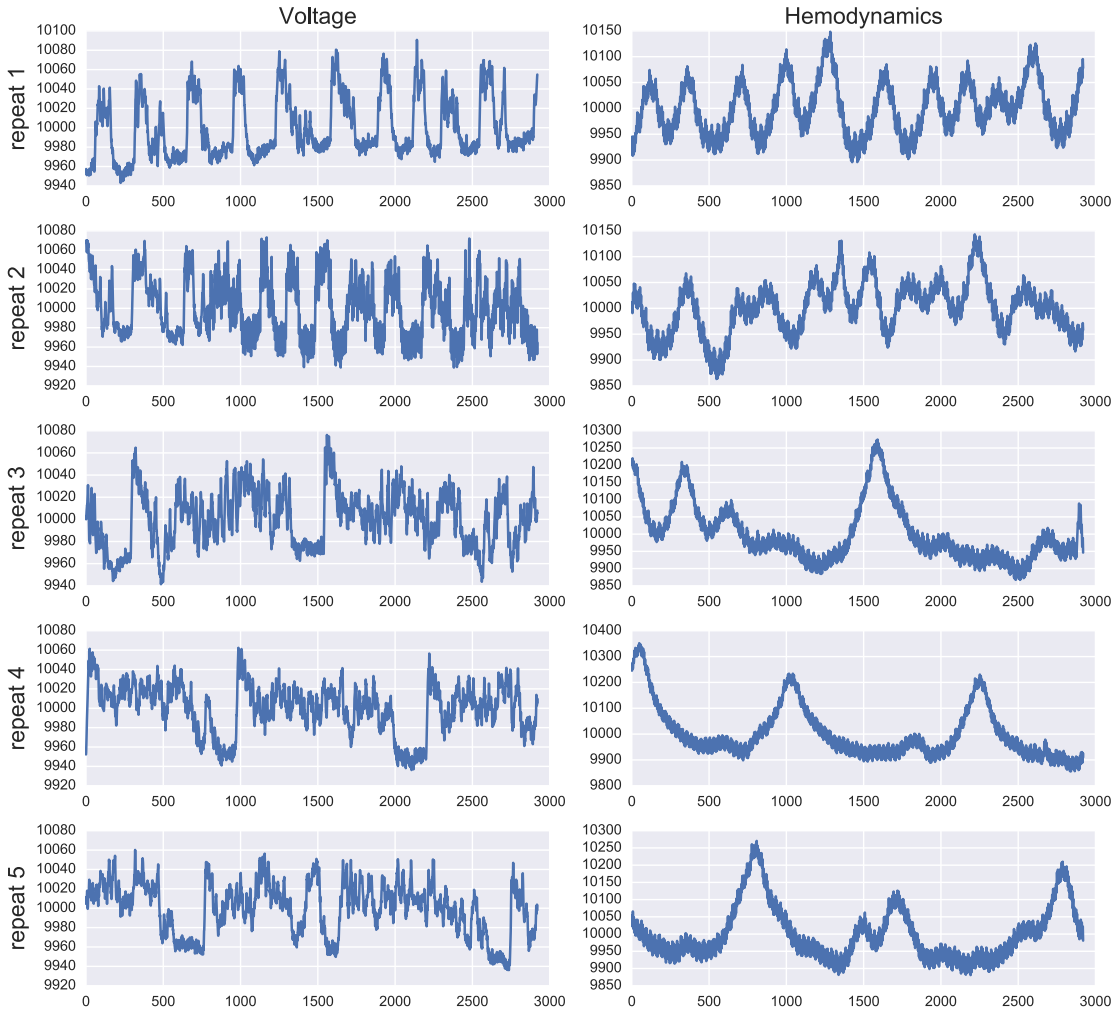
Experiment 3

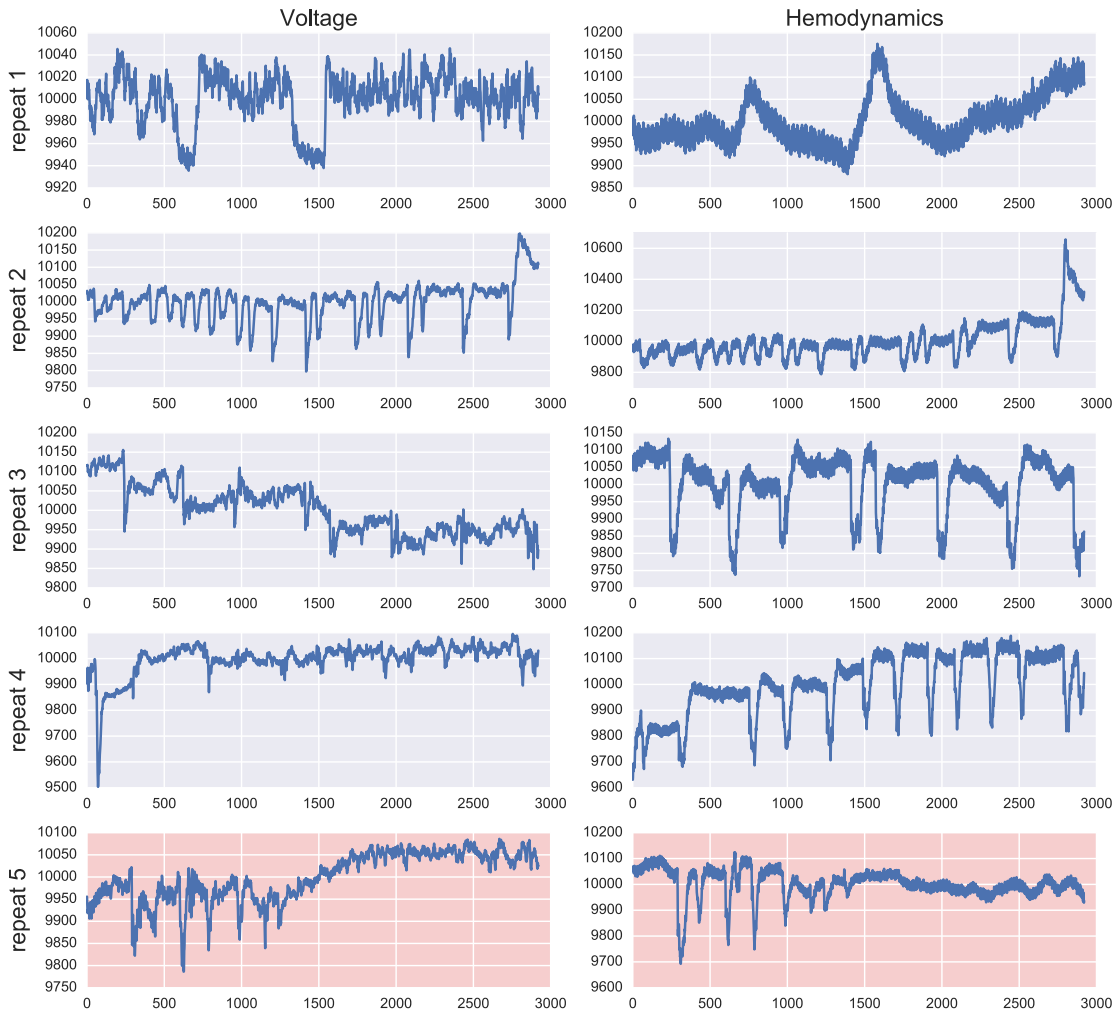


Experiment 4



Experiment 5

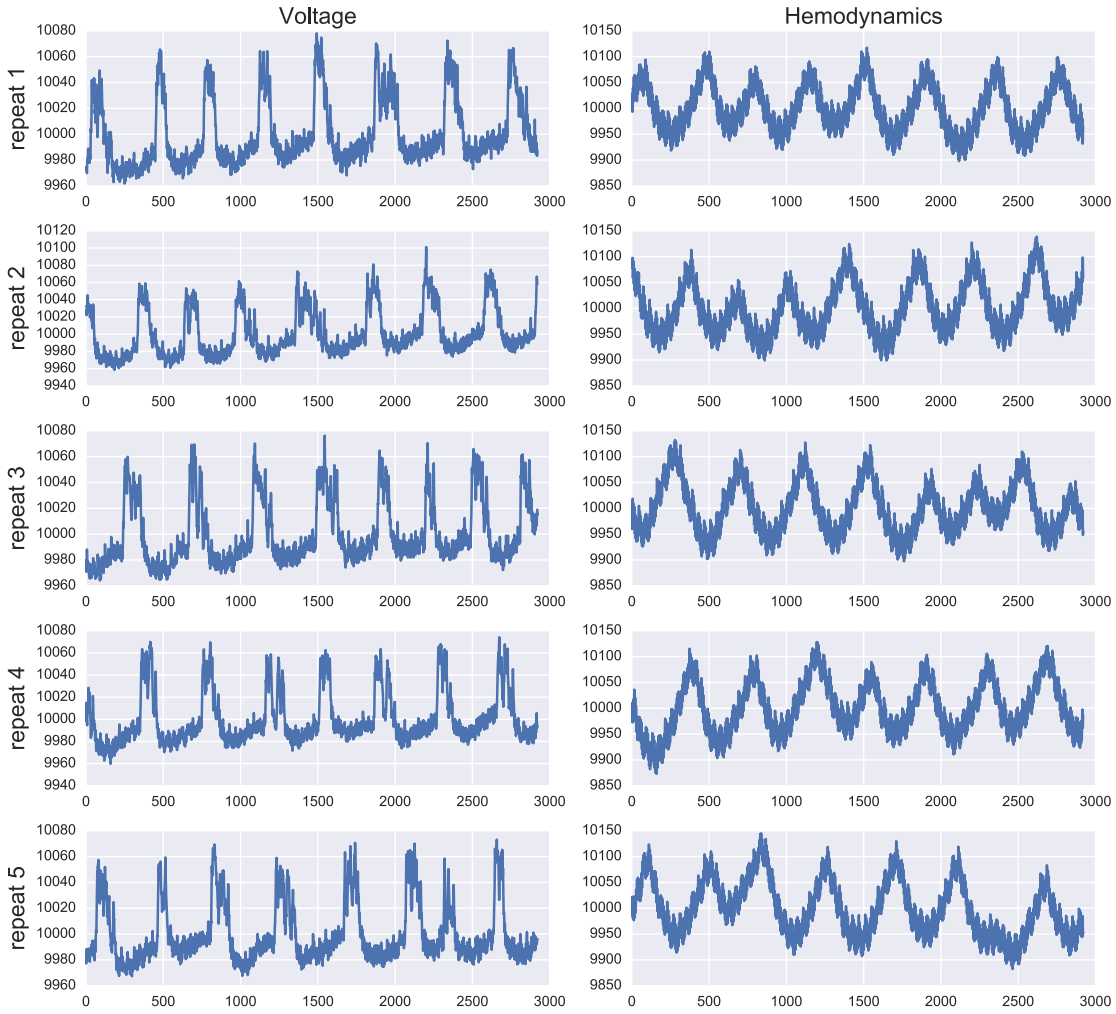




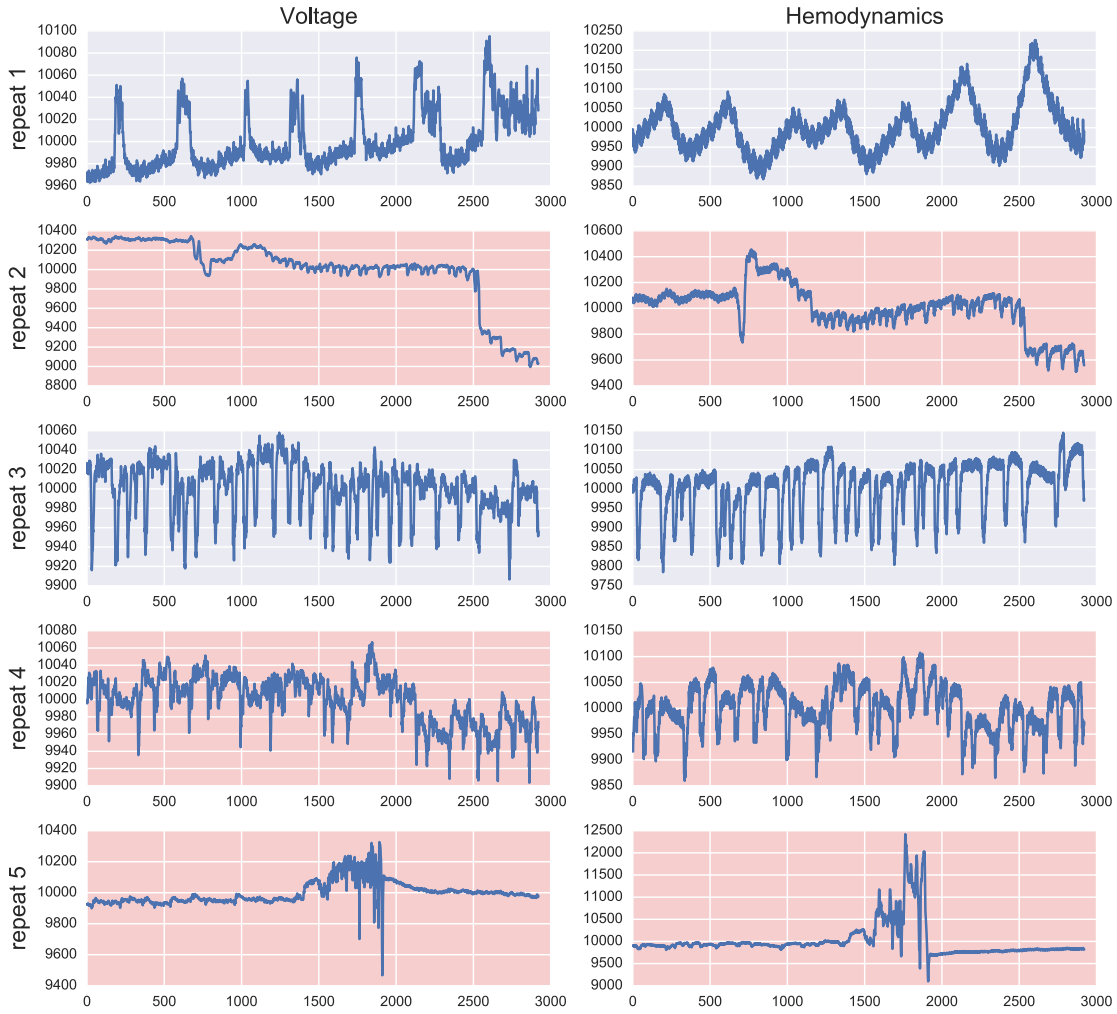
2 Mouse B

```
In [9]: for i in range(len(dataB)):
            plotData(dataB[i], i, discard['MouseB'],2)
```

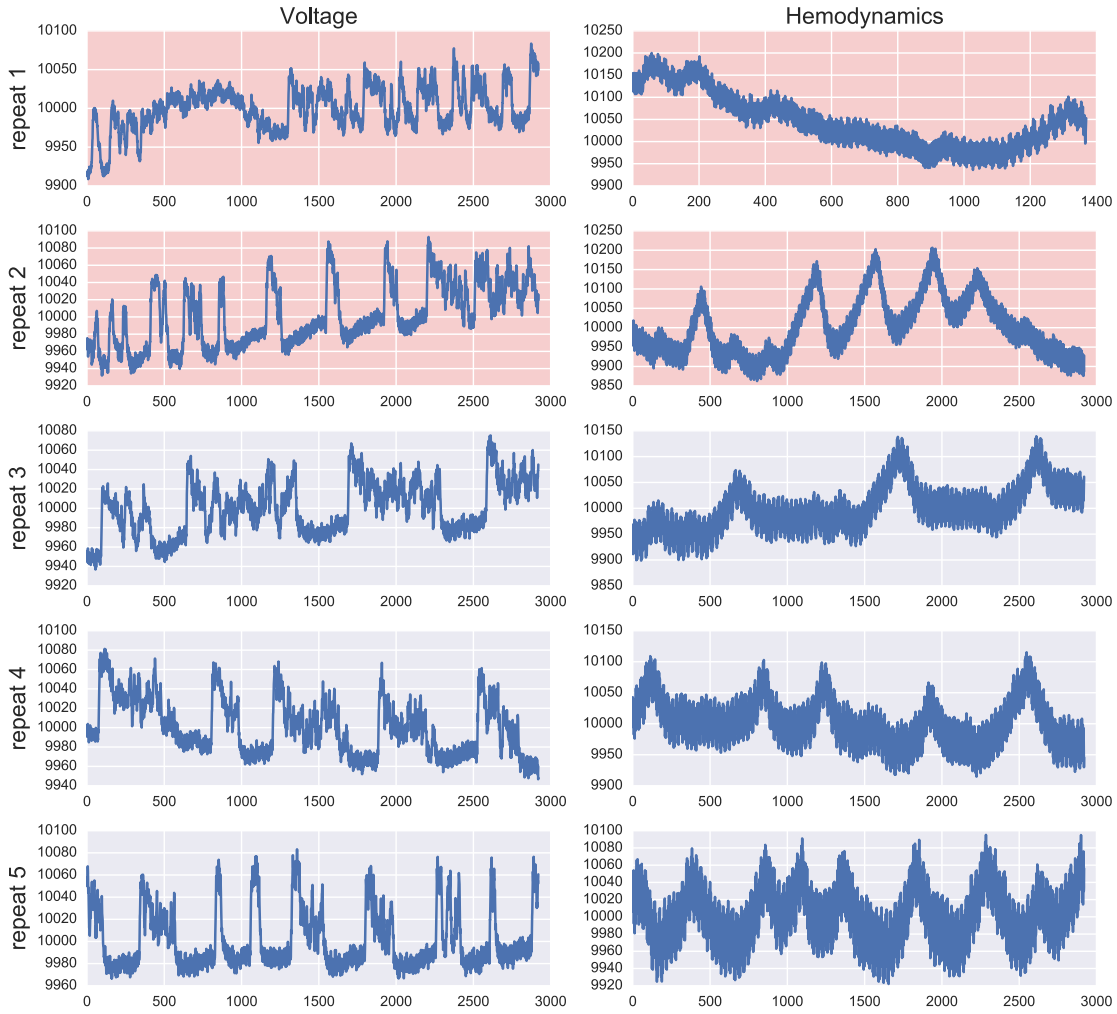
Experiment 2

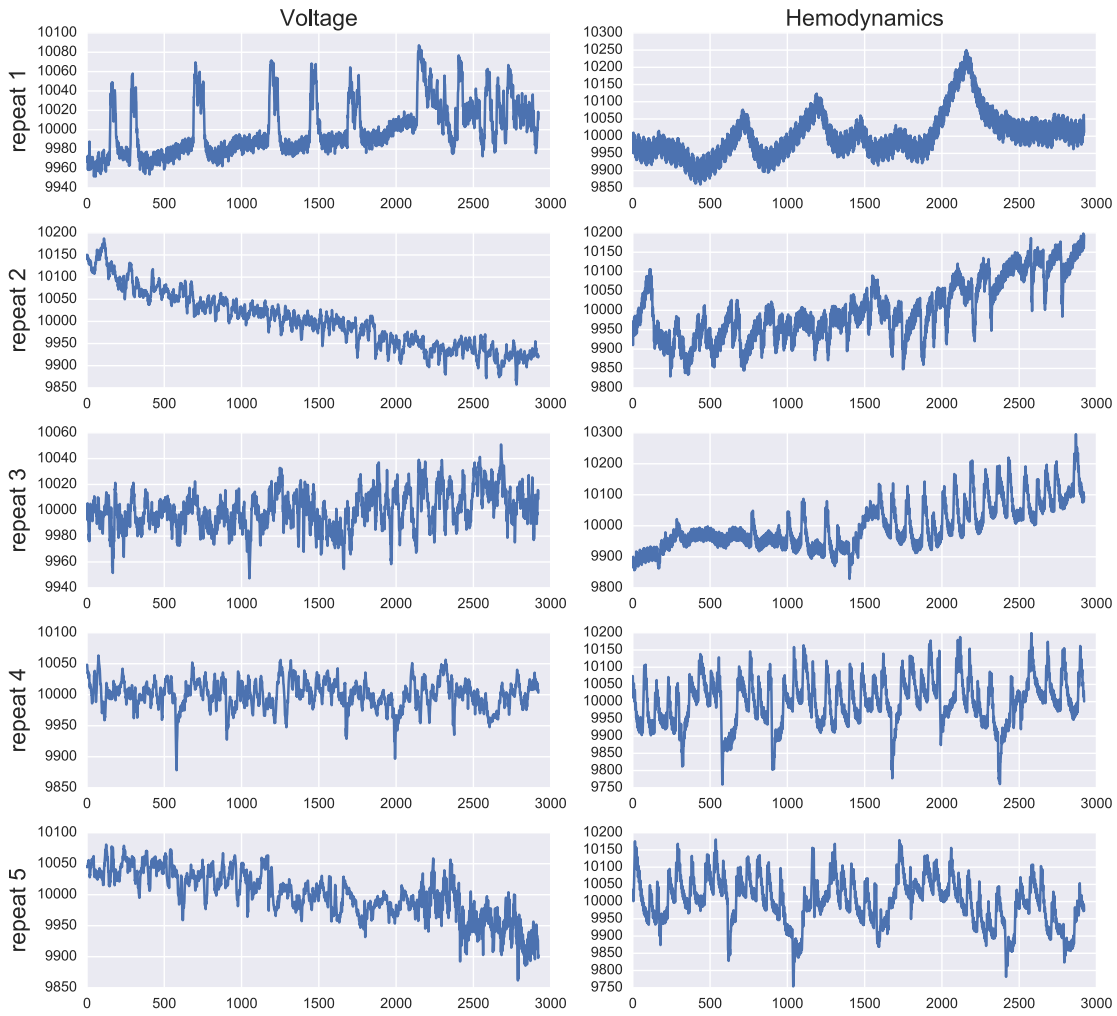


Experiment 3



Experiment 4





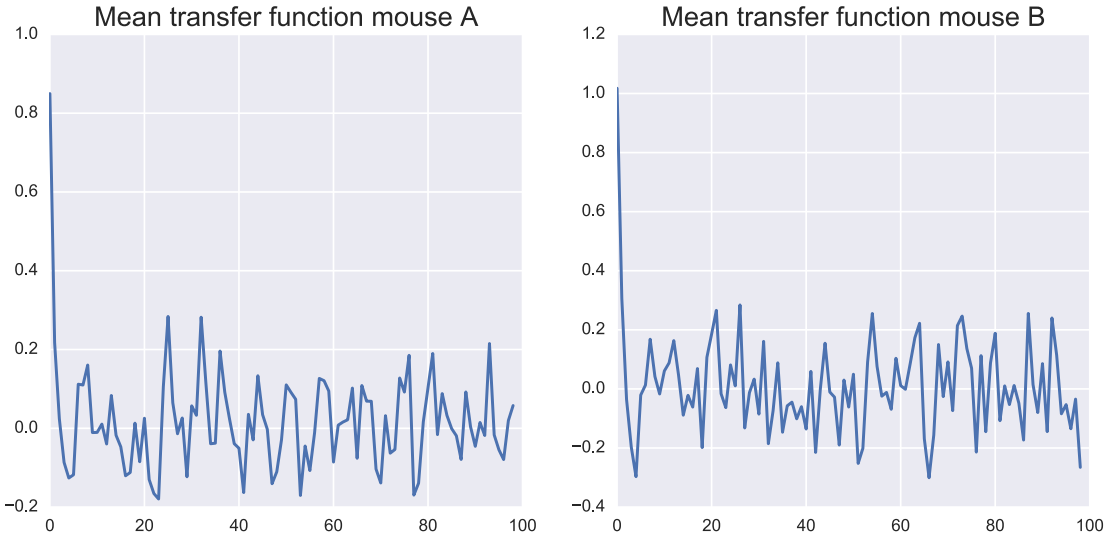
3 Transfer functions

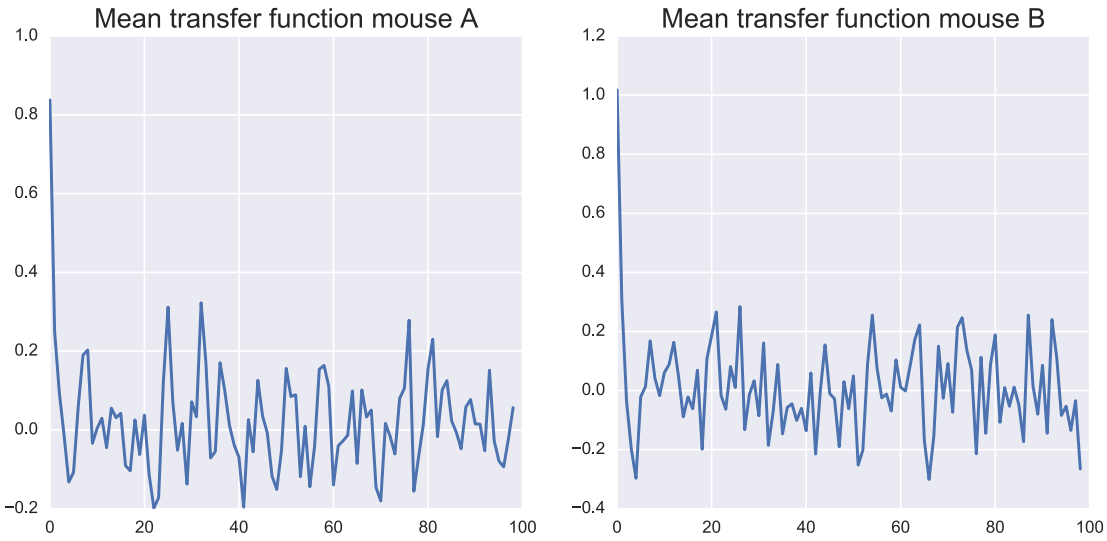
3.1 the voltage and hemodynamic signals are averaged over all pixels for the 10 following images (sliding average) and the ratio of both functions is filtered in the frequency domain.

```
In [10]: fig = plt.figure(figsize=(11,5))

         ax = fig.add_subplot(1,2,1)
         alpha = getMeanAlphaFiltered(dataA,1,100)
         ax.plot(alpha)
         ax.set_title('Mean transfer function mouse A')

         ax = fig.add_subplot(1,2,2)
```

```
alpha = getMeanAlphaFiltered(dataB,1,100)
ax.plot(alpha)
ax.set_title('Mean transfer function mouse B')


fig = plt.figure(figsize=(11,5))

ax = fig.add_subplot(1,2,1)
alpha = getMeanAlphaFiltered(goodDataA,1,100)
ax.plot(alpha)
ax.set_title('Mean transfer function mouse A')

ax = fig.add_subplot(1,2,2)
alpha = getMeanAlphaFiltered(goodDataB,1,100)
ax.plot(alpha)
ax.set_title('Mean transfer function mouse B')
```

Out[10]: <matplotlib.text.Text at 0x19bedbf90>



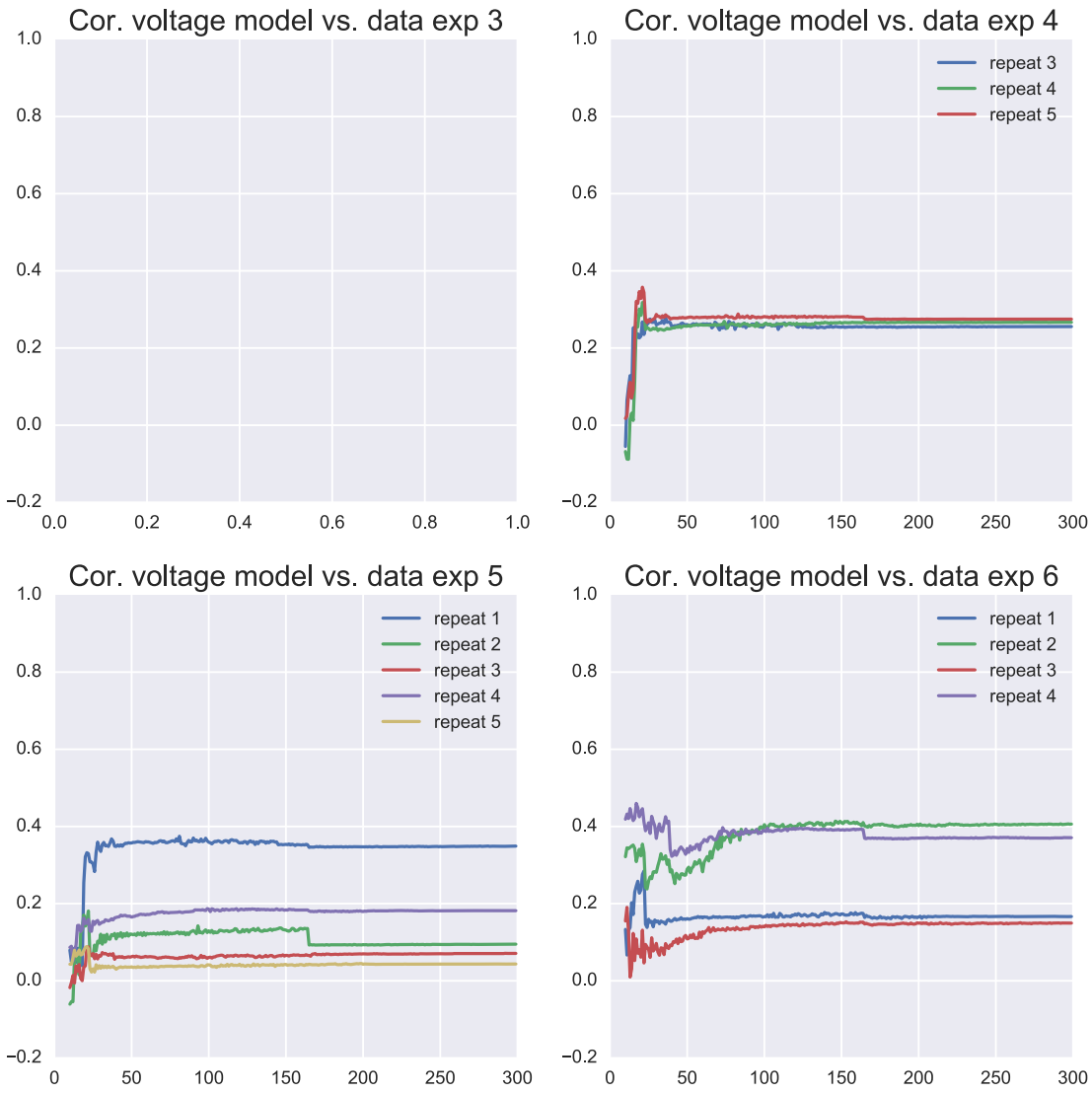


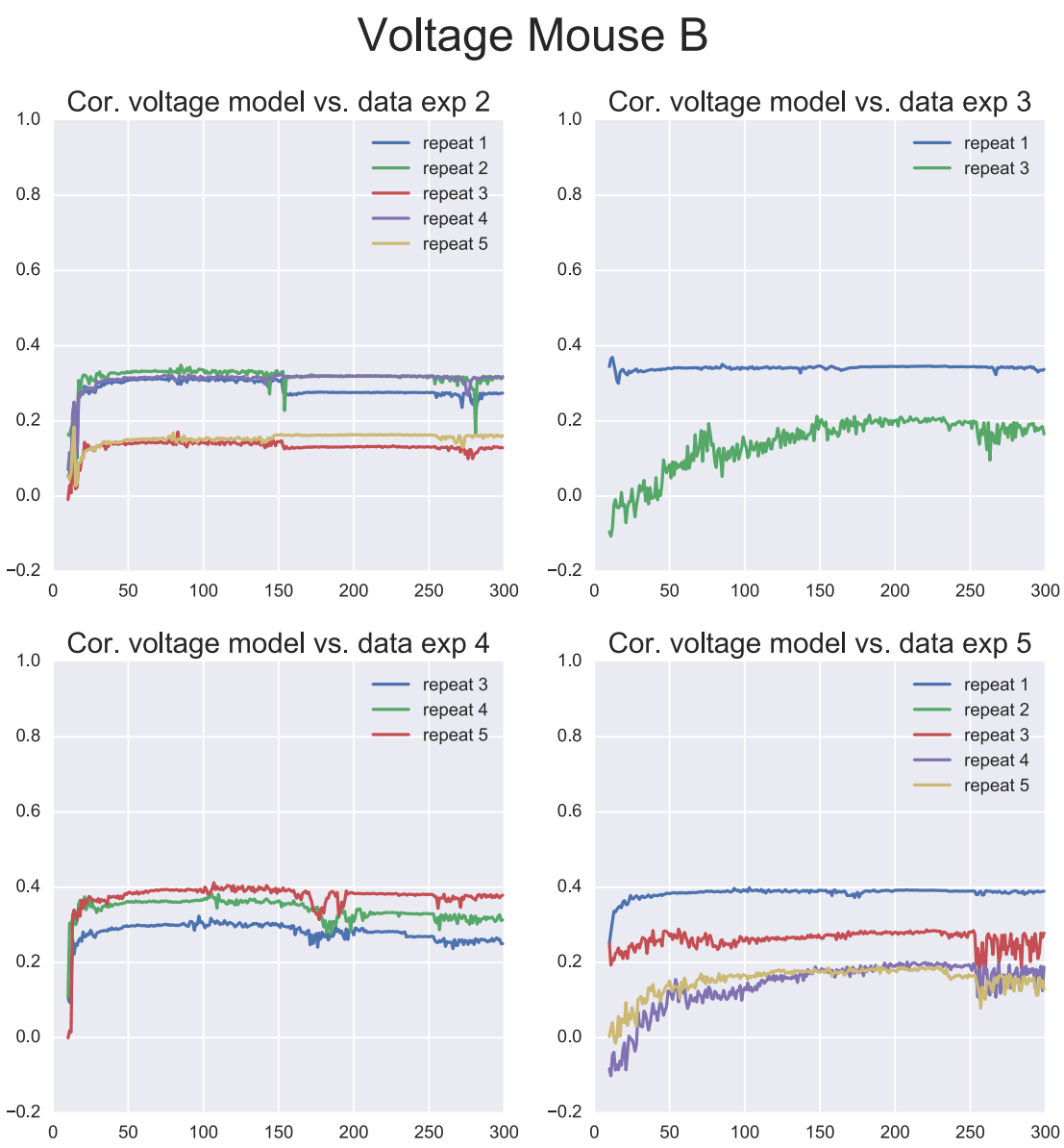
3.2 Transfer function average on all experiments and all repeats

3.2.1 Voltage

```
In [11]: # compareCorr(data[1],data[3])
         fig = plt.figure(figsize=(10,10))
         for i in range(len(dataA)):
             ax = fig.add_subplot(2,2,i+1)
             plotCorrRatioAlpha(ax, dataA[i], getMeanAlphaFiltered(goodDataA,0,300), discard['MouseA'],
         plt.suptitle('Voltage Mouse A', fontsize=26)

         fig = plt.figure(figsize=(10,10))
         for i in range(len(dataB)):
             ax = fig.add_subplot(2,2,i+1)
             plotCorrRatioAlpha(ax, dataB[i], getMeanAlphaFiltered(goodDataB,0,300), discard['MouseB'],
         plt.suptitle('Voltage Mouse B', fontsize=26)

Out[11]: <matplotlib.text.Text at 0x16c7fbbd0>
```

Voltage Mouse A

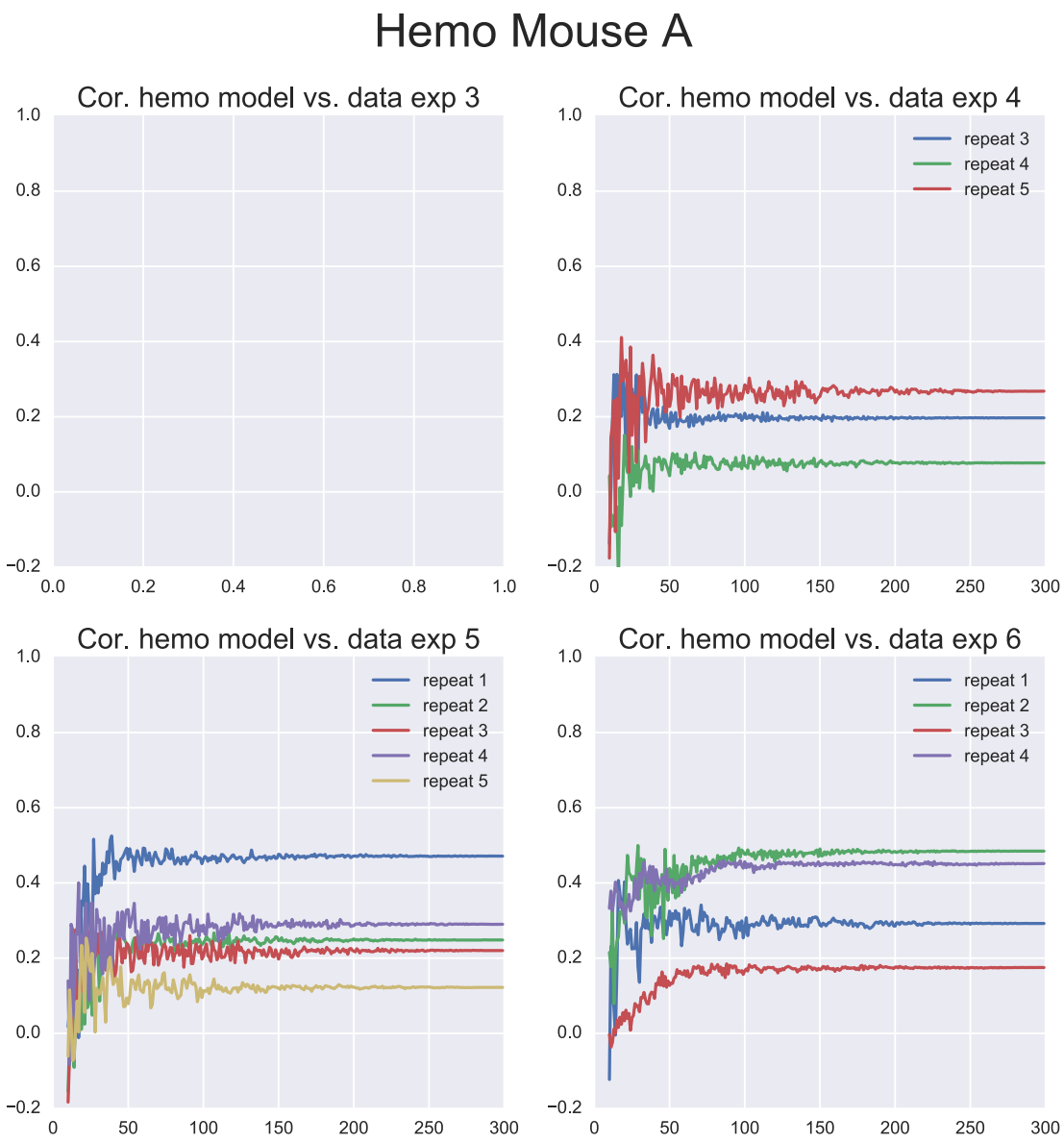




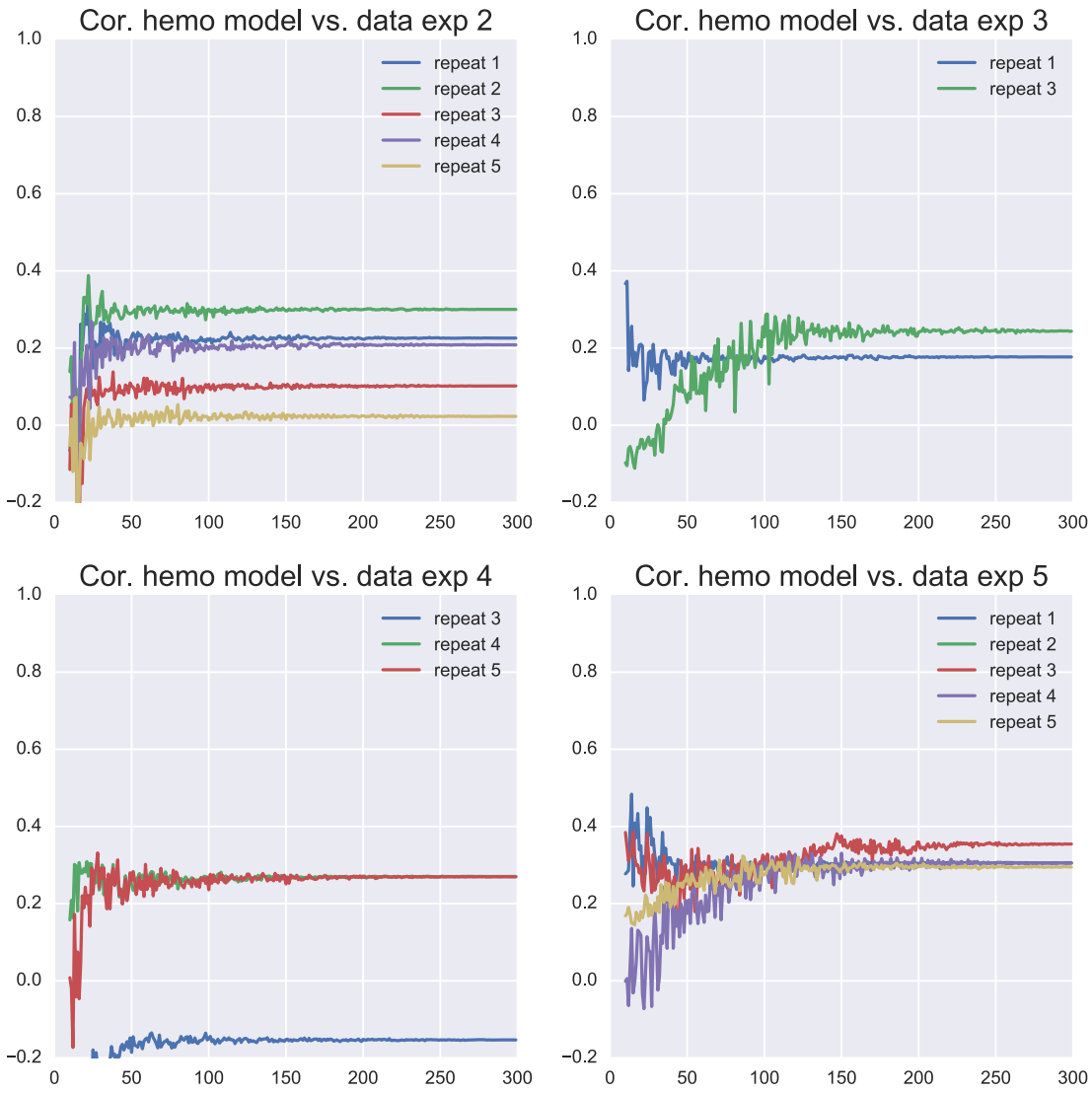
3.2.2 Hemo

```
In [12]: fig = plt.figure(figsize=(10,10))
         for i in range(len(dataA)):
             ax = fig.add_subplot(2,2,i+1)
             plotCorrHemoAlpha(ax, dataA[i], getMeanAlphaFiltered(goodDataA,0,100), discard['MouseA'], c
         plt.suptitle('Hemo Mouse A', fontsize=26)

         fig = plt.figure(figsize=(10,10))
         for i in range(len(dataB)):
             ax = fig.add_subplot(2,2,i+1)
             plotCorrHemoAlpha(ax, dataB[i], getMeanAlphaFiltered(goodDataB,0,100), discard['MouseB'], c
         plt.suptitle('Hemo Mouse B', fontsize=26)
```

Out[12]: <matplotlib.text.Text at 0x117ab6410>



Hemo Mouse B



In [13]:
```
## With the 'bad' data
```

In [14]:
```
# # compareCorr(data[1],data[3])
# fig = plt.figure(figsize=(10,10))
# for i in range(len(dataA)):
#     ax = fig.add_subplot(2,2,i+1)
#     plotCorrRatioAlpha(ax, dataA[i], getMeanAlphaFiltered(dataA,0,300), exp = i, start = 3)
```

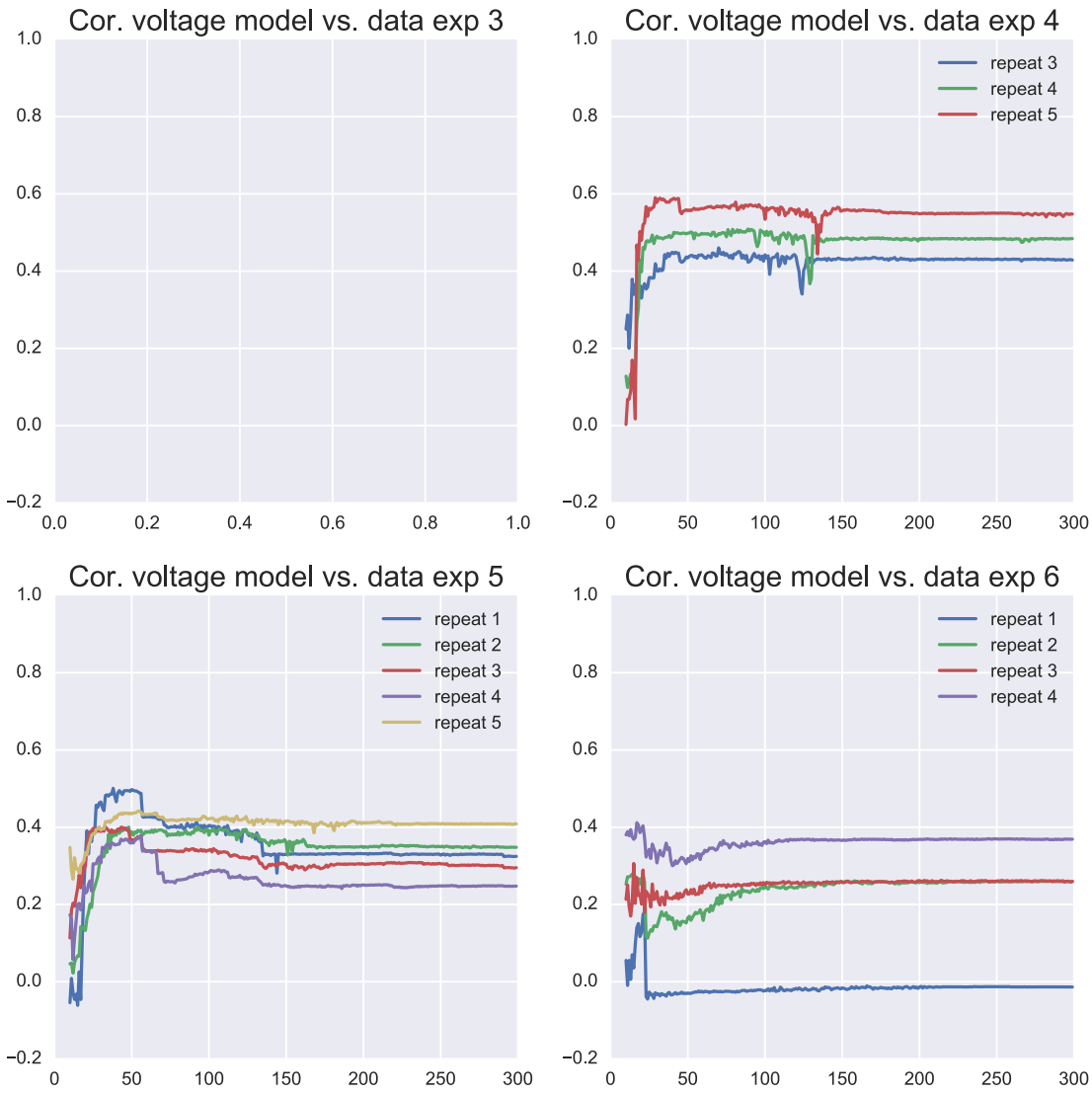
3.3 Transfer function average through all repeats, on each experiment separately (overfitting?)

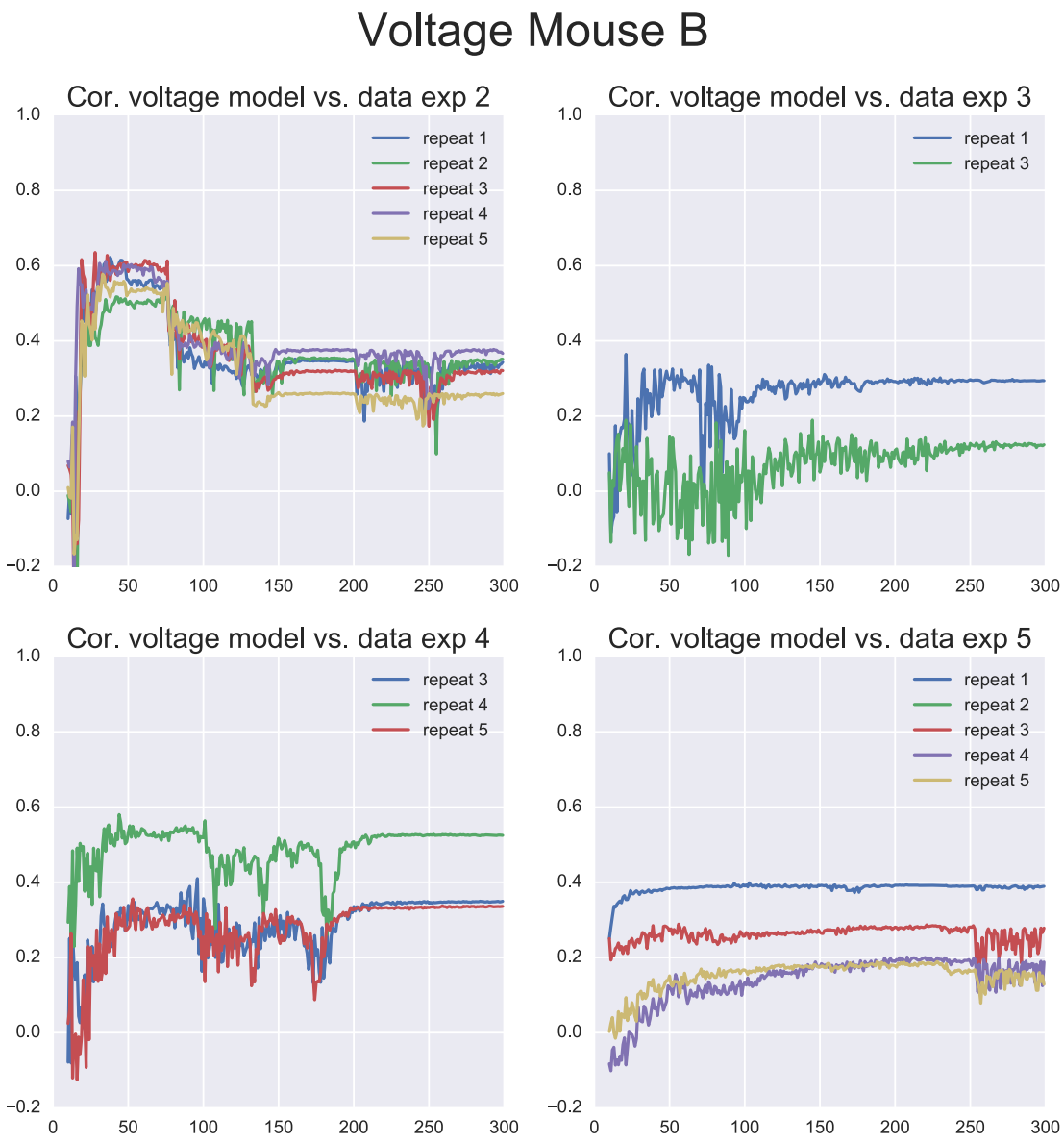
3.3.1 Voltage

In [15]:
```
fig = plt.figure(figsize=(10,10))
for i in range(len(dataA)):
    ax = fig.add_subplot(2,2,i+1)
    plotCorrRatioAlpha(ax, dataA[i], getMeanAlphaFiltered([dataA[i]],0,300), discard['MouseA']
plt.suptitle('Voltage Mouse A', fontsize=26)

fig = plt.figure(figsize=(10,10))
for i in range(len(dataB)):
    ax = fig.add_subplot(2,2,i+1)
    plotCorrRatioAlpha(ax, dataB[i], getMeanAlphaFiltered([dataB[i]],0,300), discard['MouseB']
plt.suptitle('Voltage Mouse B', fontsize=26)
```

Out[15]:
```
<matplotlib.text.Text at 0x19d6ba590>
```

Voltage Mouse A

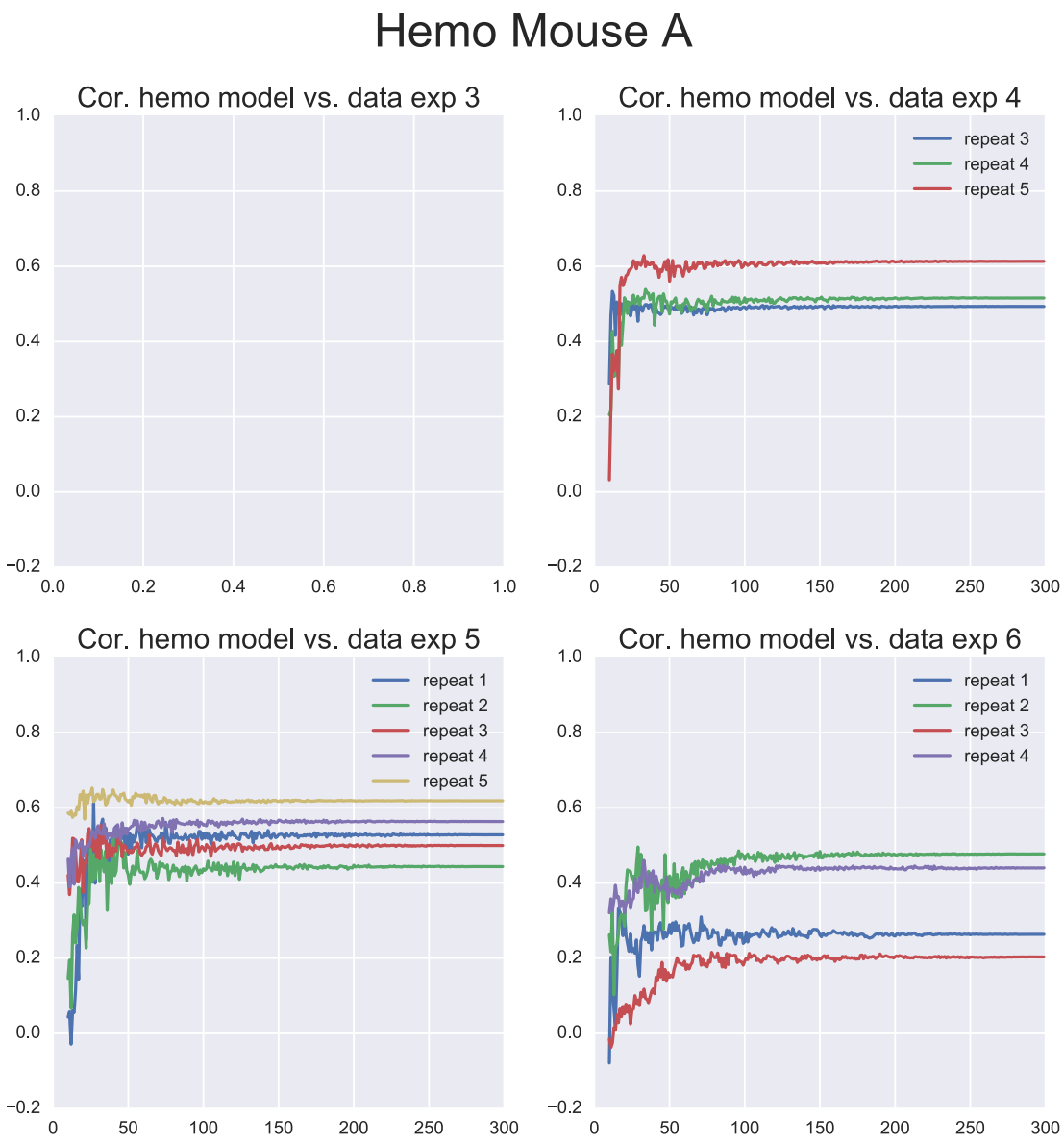


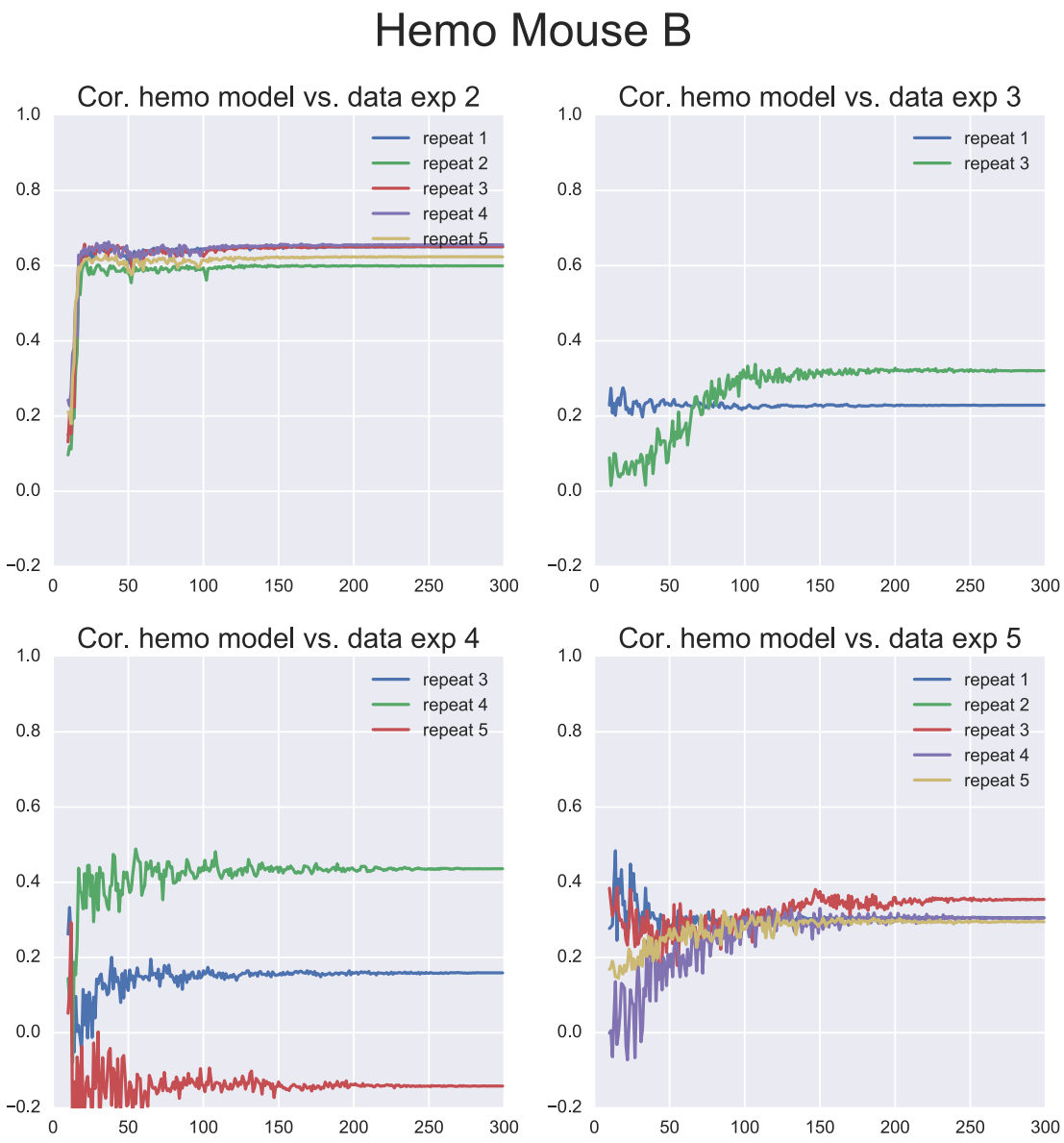


3.3.2 Hemo

```
In [16]: fig = plt.figure(figsize=(10,10))
         for i in range(len(dataA)):
             ax = fig.add_subplot(2,2,i+1)
             plotCorrHemoAlpha(ax, dataA[i], getMeanAlphaFiltered([dataA[i]],0,100), discard['MouseA'],
         plt.suptitle('Hemo Mouse A', fontsize=26)

         fig = plt.figure(figsize=(10,10))
         for i in range(len(dataB)):
             ax = fig.add_subplot(2,2,i+1)
             plotCorrHemoAlpha(ax, dataB[i], getMeanAlphaFiltered([dataB[i]],0,100), discard['MouseB'],
         plt.suptitle('Hemo Mouse B', fontsize=26)
```

Out[16]: <matplotlib.text.Text at 0x1b83a4610>





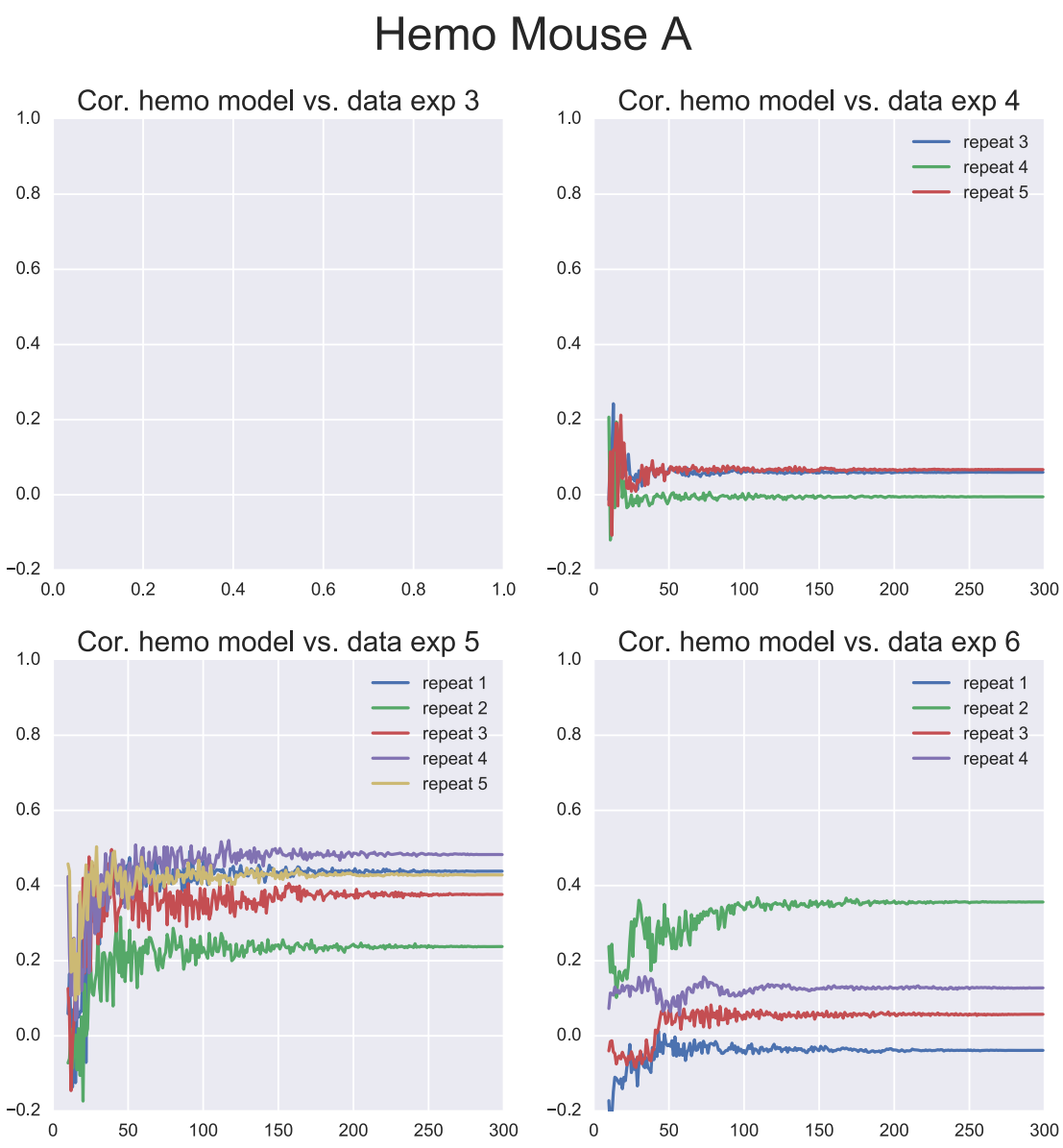
3.3.3 Using transfer function from mouse B to model mouse A's hemo signal

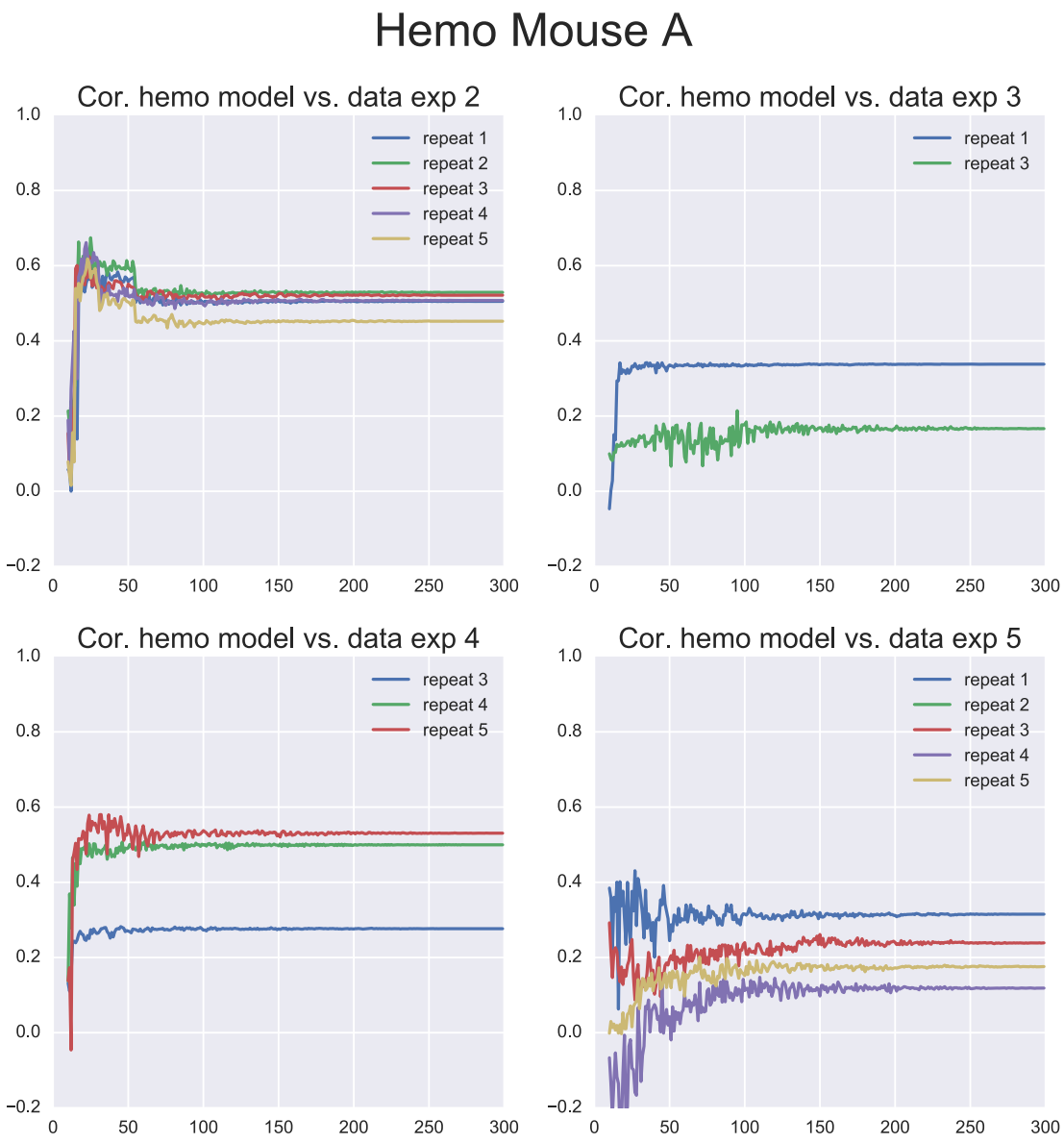
3.3.4 Using transfer function from mouse A to model mouse B's hemo signal

```
In [17]: fig = plt.figure(figsize=(10,10))
         for i in range(len(dataA)):
             ax = fig.add_subplot(2,2,i+1)
             plotCorrHemoAlpha(ax, dataA[i], getMeanAlphaFiltered([dataB[i]],0,100), discard['MouseA'],
         plt.suptitle('Hemo Mouse A', fontsize=26)

         fig = plt.figure(figsize=(10,10))
         for i in range(len(dataB)):
             ax = fig.add_subplot(2,2,i+1)
```

```
        plotCorrHemoAlpha(ax, dataB[i], getMeanAlphaFiltered([dataA[i]],0,100), discard['MouseB'],
    plt.suptitle('Hemo Mouse A', fontsize=26)
```

Out[17]: <matplotlib.text.Text at 0x16c7ba4d0>





3.4 We choose 100Hz as maximum frequency to low pass filter the data

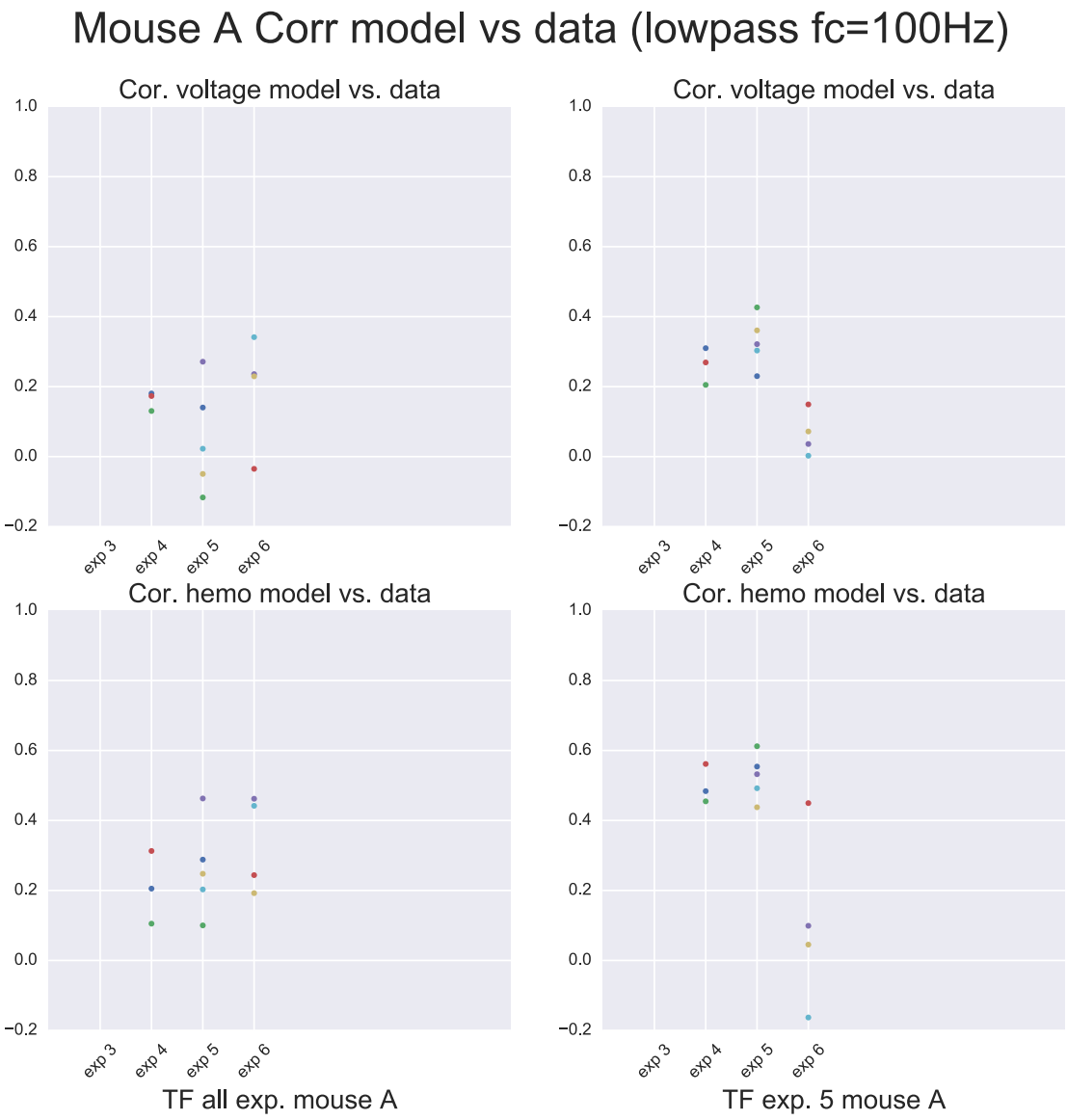
3.5 We compare using the transfer function (TF) obtained for all the experiments with the one from experiment 5 of mouse A which seems to have good data

3.5.1 Mouse A

```
In [18]: fig = plt.figure(figsize=(11,10))
         ax = fig.add_subplot(221)
         plotCorrR(ax, dataA, getMeanAlphaFiltered(dataA,0,100), discard['MouseA'], 100, 3)
         ax = fig.add_subplot(222)
         plotCorrR(ax, dataA, getMeanAlphaFiltered([dataA[2]],0,100), discard['MouseA'], 100, 3)
```

```
ax = fig.add_subplot(223)
plotCorrH(ax, dataA, getMeanAlphaFiltered(dataA,0,100), discard['MouseA'], 100, 3)
ax.set_xlabel('TF all exp. mouse A')
ax = fig.add_subplot(224)
plotCorrH(ax, dataA, getMeanAlphaFiltered([dataA[2]],0,100), discard['MouseA'], 100, 3)
ax.set_xlabel('TF exp. 5 mouse A')
plt.suptitle('Mouse A Corr model vs data (lowpass fc=100Hz)', fontsize=26)
```

Out[18]: <matplotlib.text.Text at 0x15b8b2fd0>

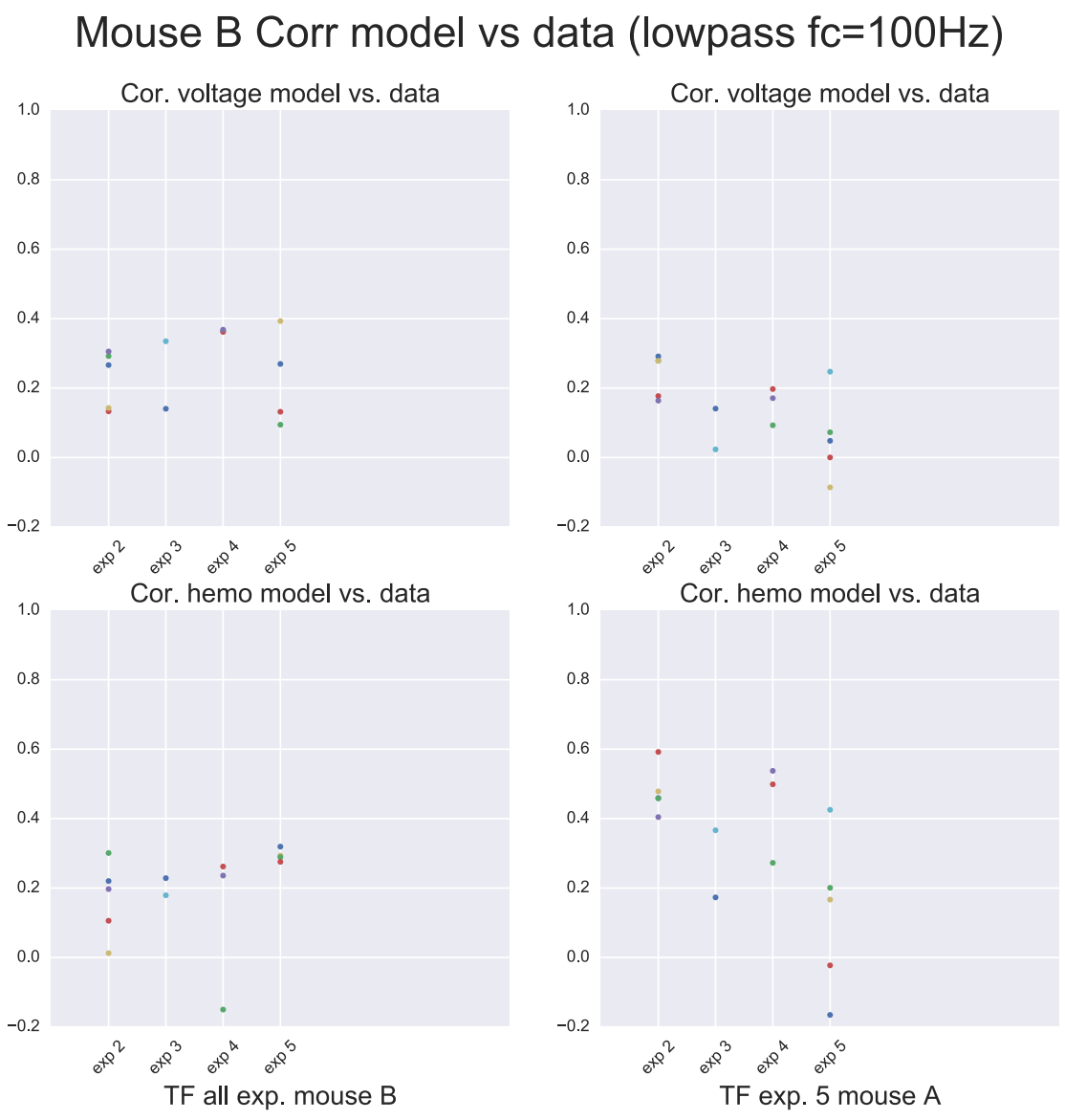


3.5.2 Mouse B

```
In [19]: fig = plt.figure(figsize=(11,10))
         ax = fig.add_subplot(221)
```

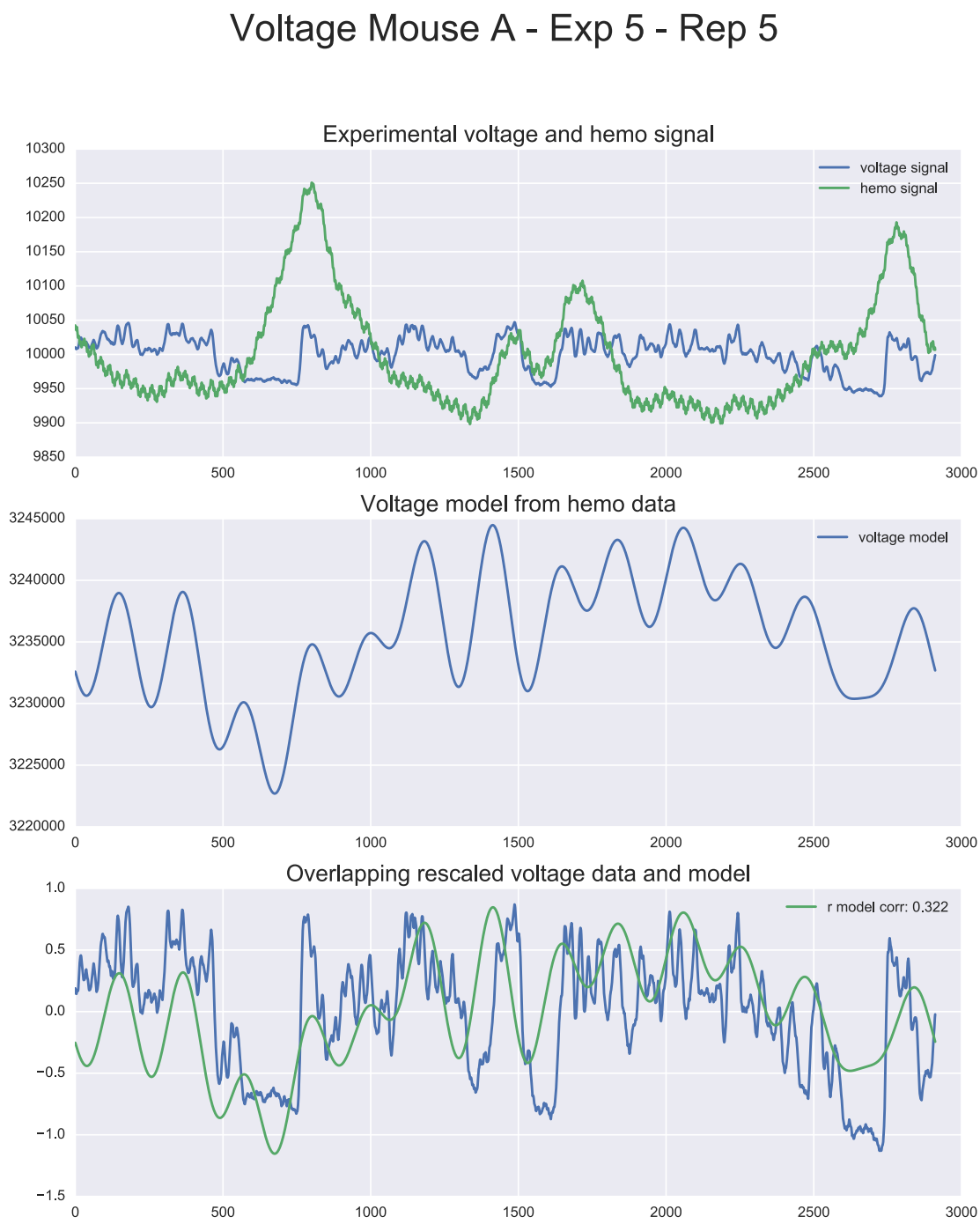
```
plotCorrR(ax, dataB, getMeanAlphaFiltered(dataB,0,100), discard['MouseB'], 100, 2)
ax = fig.add_subplot(222)
plotCorrR(ax, dataB, getMeanAlphaFiltered([dataA[2]],0,100), discard['MouseB'], 100, 2)
ax = fig.add_subplot(223)
plotCorrH(ax, dataB, getMeanAlphaFiltered(dataB,0,100), discard['MouseB'], 100, 2)
ax.set_xlabel('TF all exp. mouse B')
ax = fig.add_subplot(224)
plotCorrH(ax, dataB, getMeanAlphaFiltered([dataA[2]],0,100), discard['MouseB'], 100, 2)
ax.set_xlabel('TF exp. 5 mouse A')
plt.suptitle('Mouse B Corr model vs data (lowpass fc=100Hz)', fontsize=26)
```

Out[19]: <matplotlib.text.Text at 0x19d24be50>

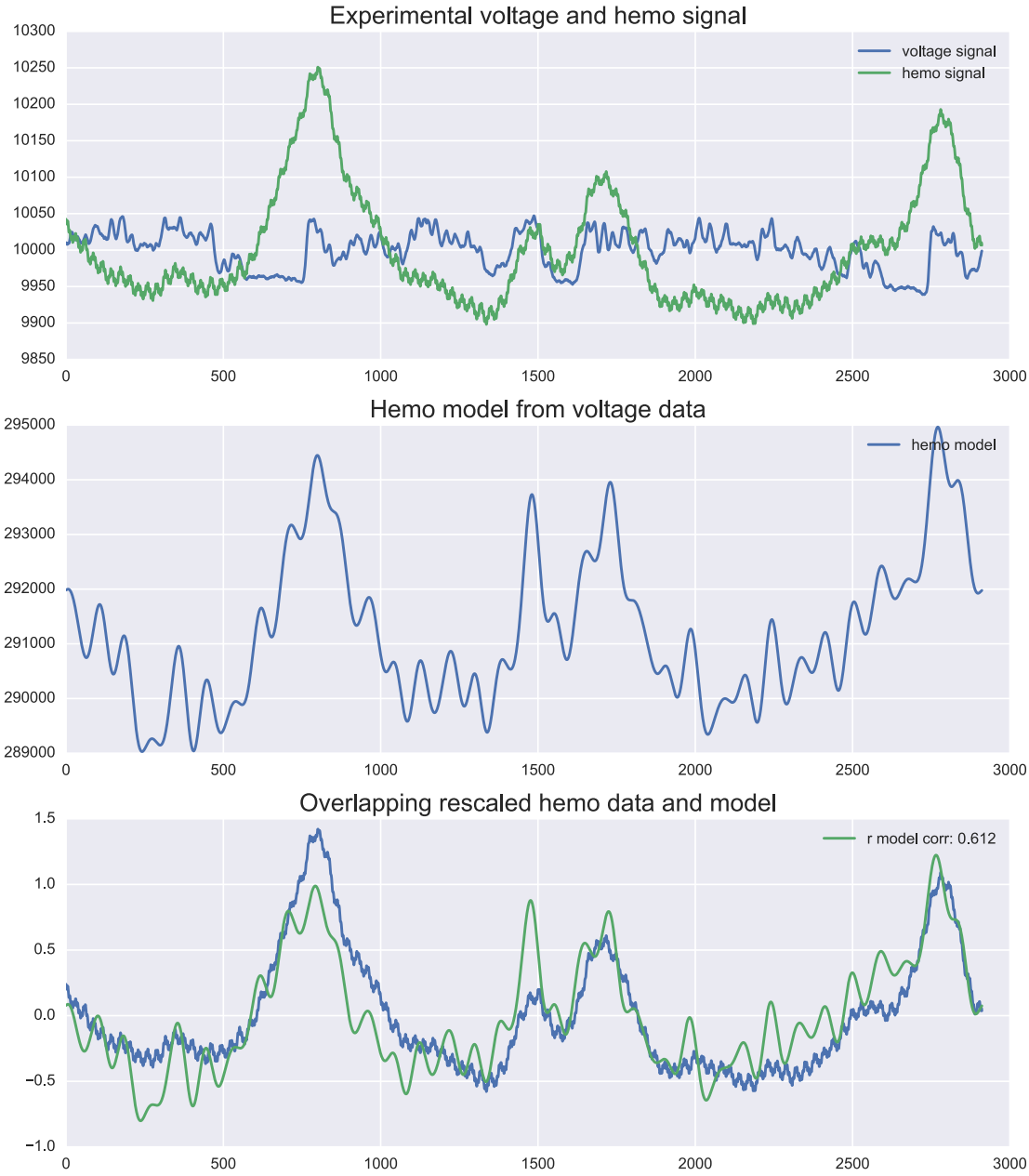


3.6 For the following plots, I took the transfer function obtain from experiment 5 of mouse A and applied to differents repeats/experiments for mouse A and B

```
In [20]: plotRModel(dataA, mouse = 'A', exp = 2, repeat = 4,alpha = getMeanAlphaFiltered([dataA[2]],0,1
         plotHModel(dataA, mouse = 'A', exp = 2, repeat = 4,alpha = getMeanAlphaFiltered([dataA[2]],0,1
```



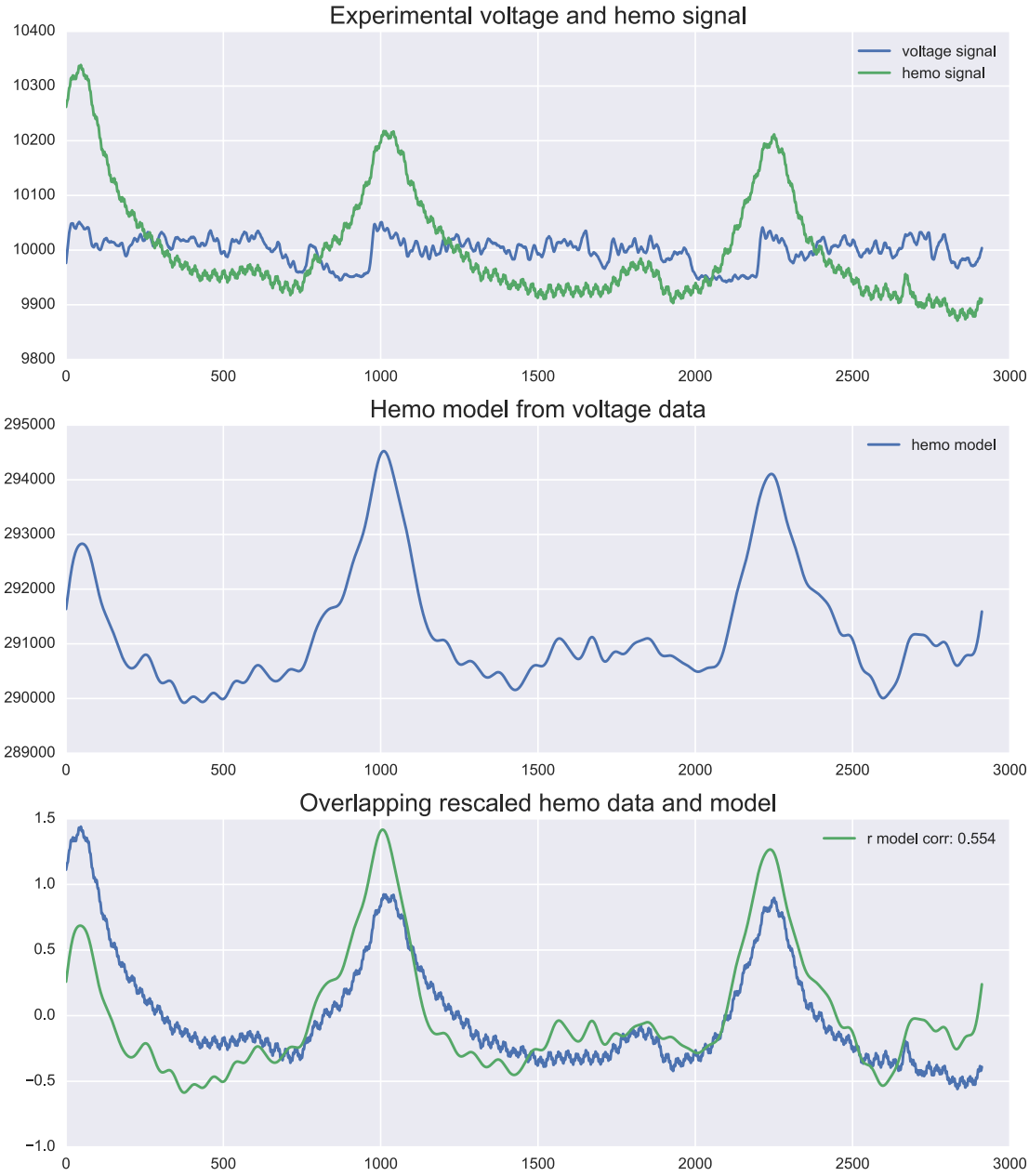
Hemo. Mouse A - Exp 5 - Rep 5



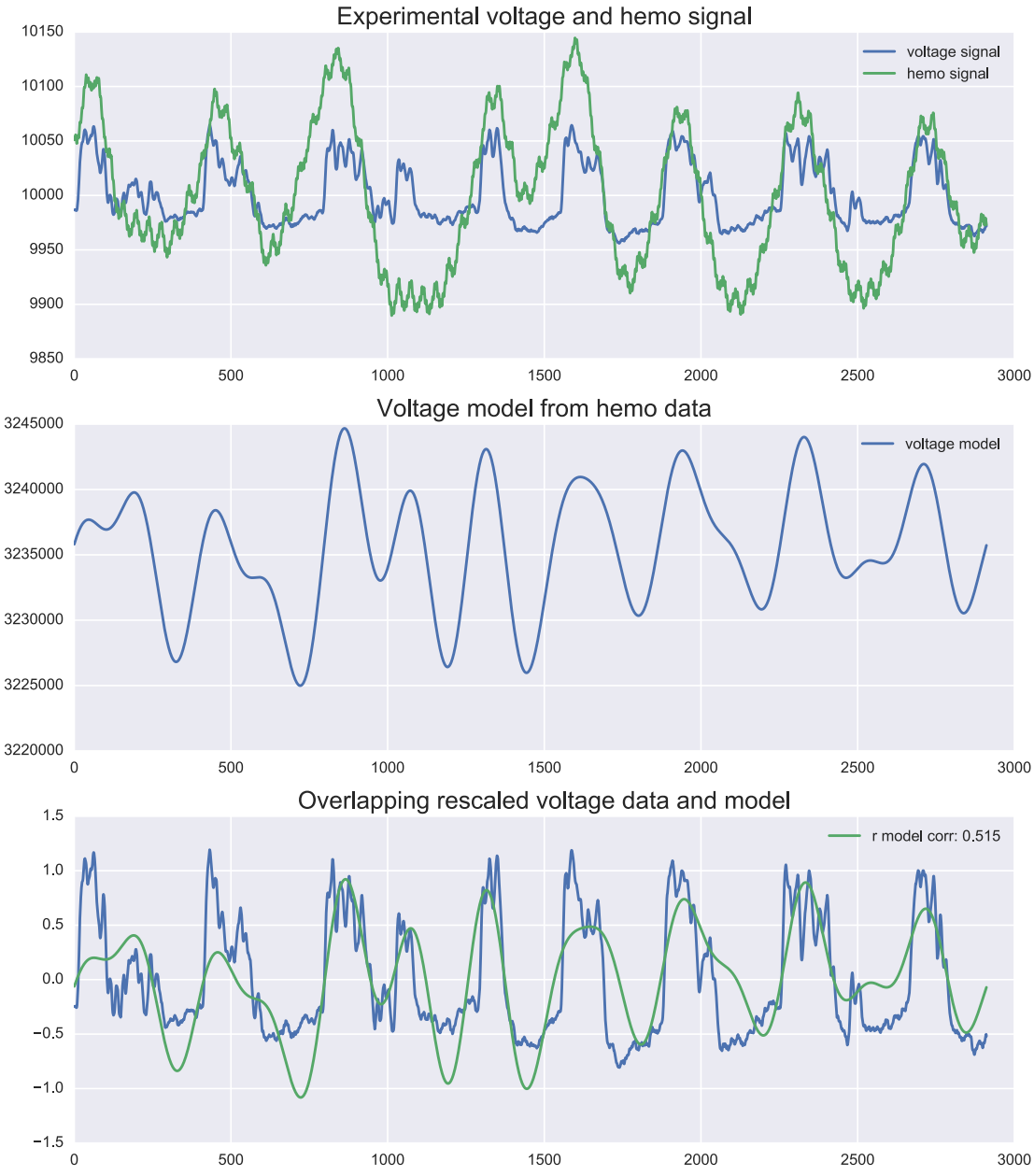

```
In [21]: plotRModel(dataA, mouse = 'A', exp = 2, repeat = 3,alpha = getMeanAlphaFiltered([dataA[2]],0,1
         plotHModel(dataA, mouse = 'A', exp = 2, repeat = 3,alpha = getMeanAlphaFiltered([dataA[2]],0,1

         plotRModel(dataA, mouse = 'A', exp = 1, repeat = 4,alpha = getMeanAlphaFiltered([dataA[2]],0,1
         plotHModel(dataA, mouse = 'A', exp = 1, repeat = 4,alpha = getMeanAlphaFiltered([dataA[2]],0,1
```

Voltage Mouse A - Exp 5 - Rep 4

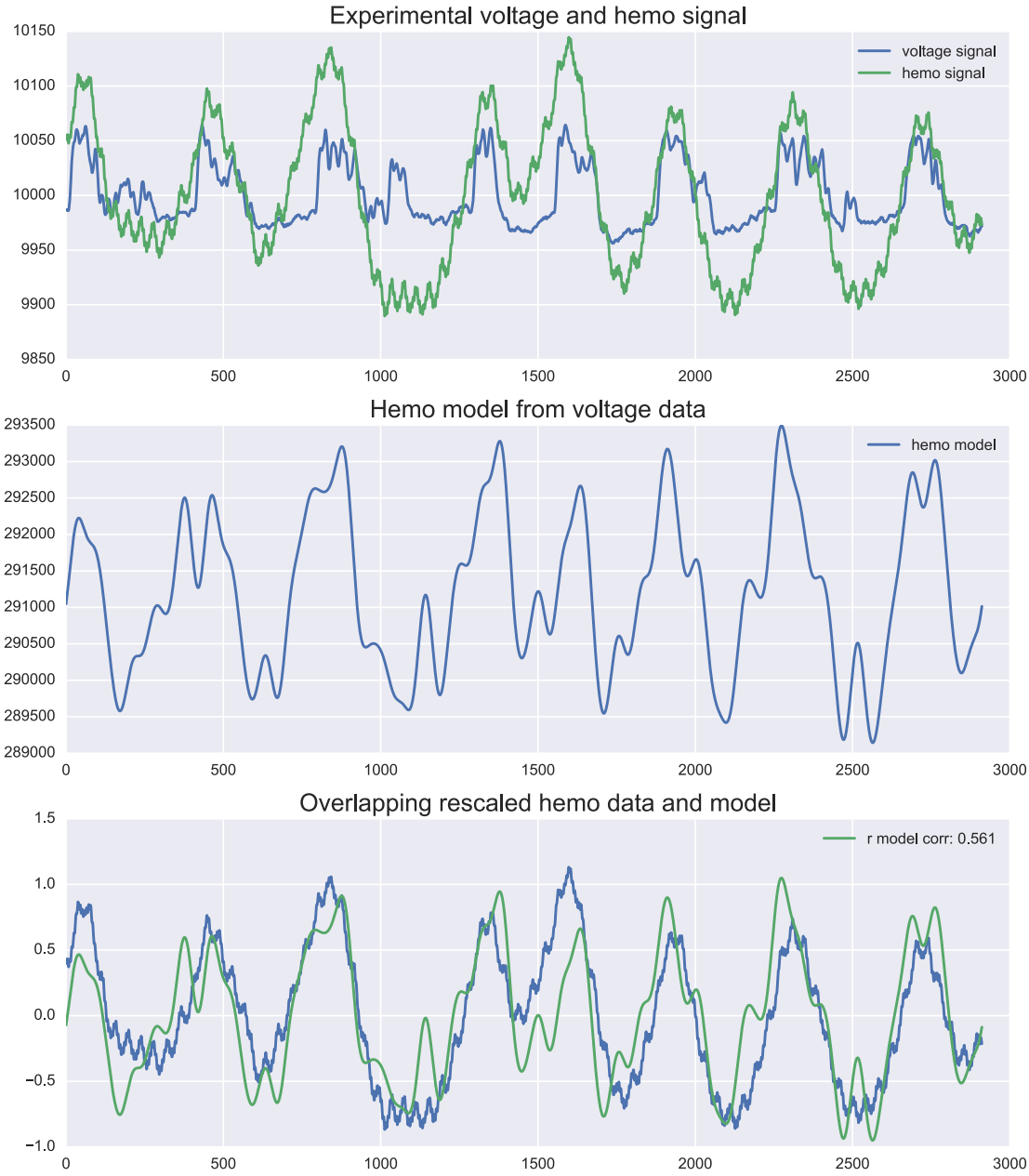
Hemo. Mouse A - Exp 5 - Rep 4



Voltage Mouse A - Exp 4 - Rep 5

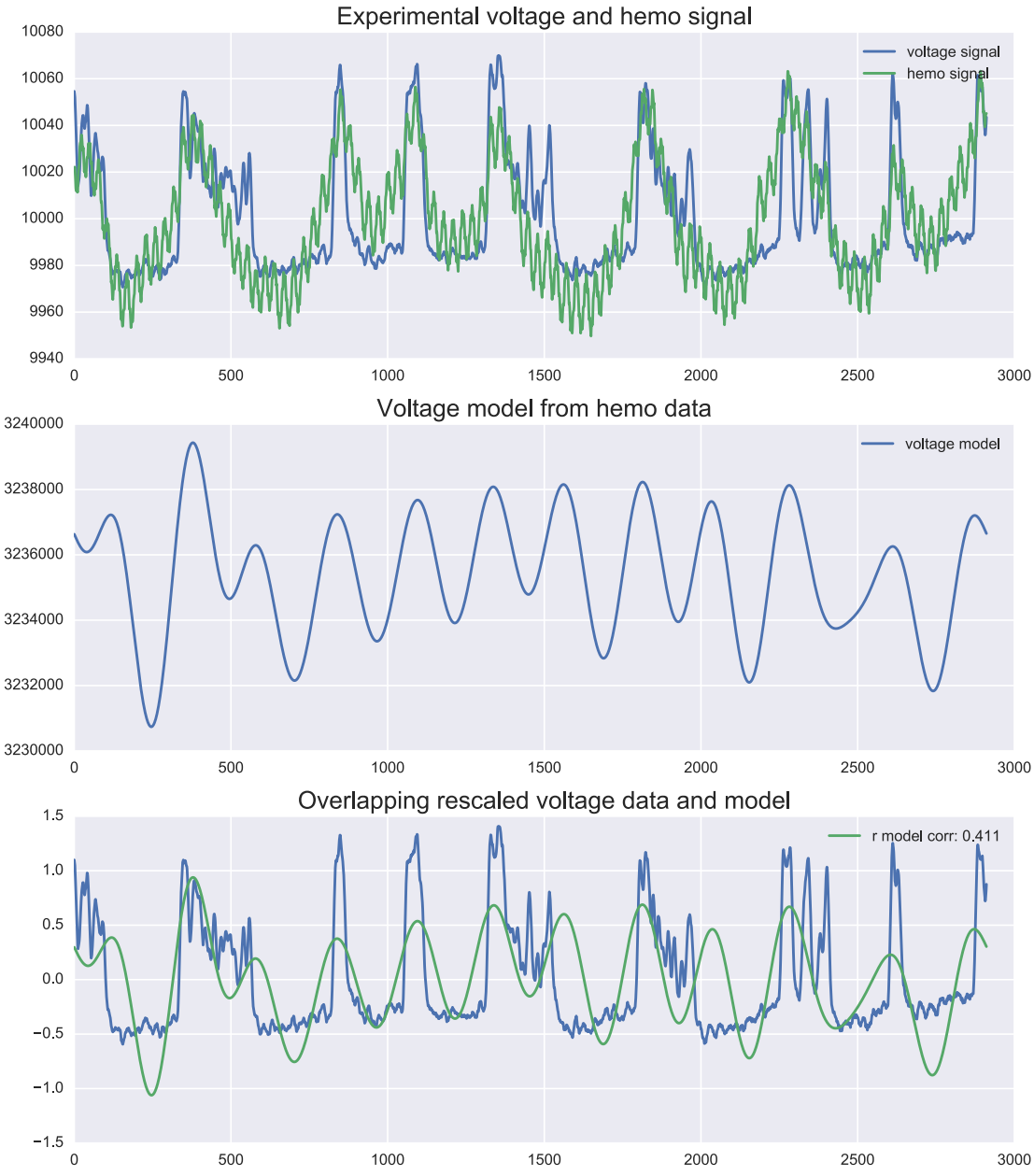


Hemo. Mouse A - Exp 4 - Rep 5

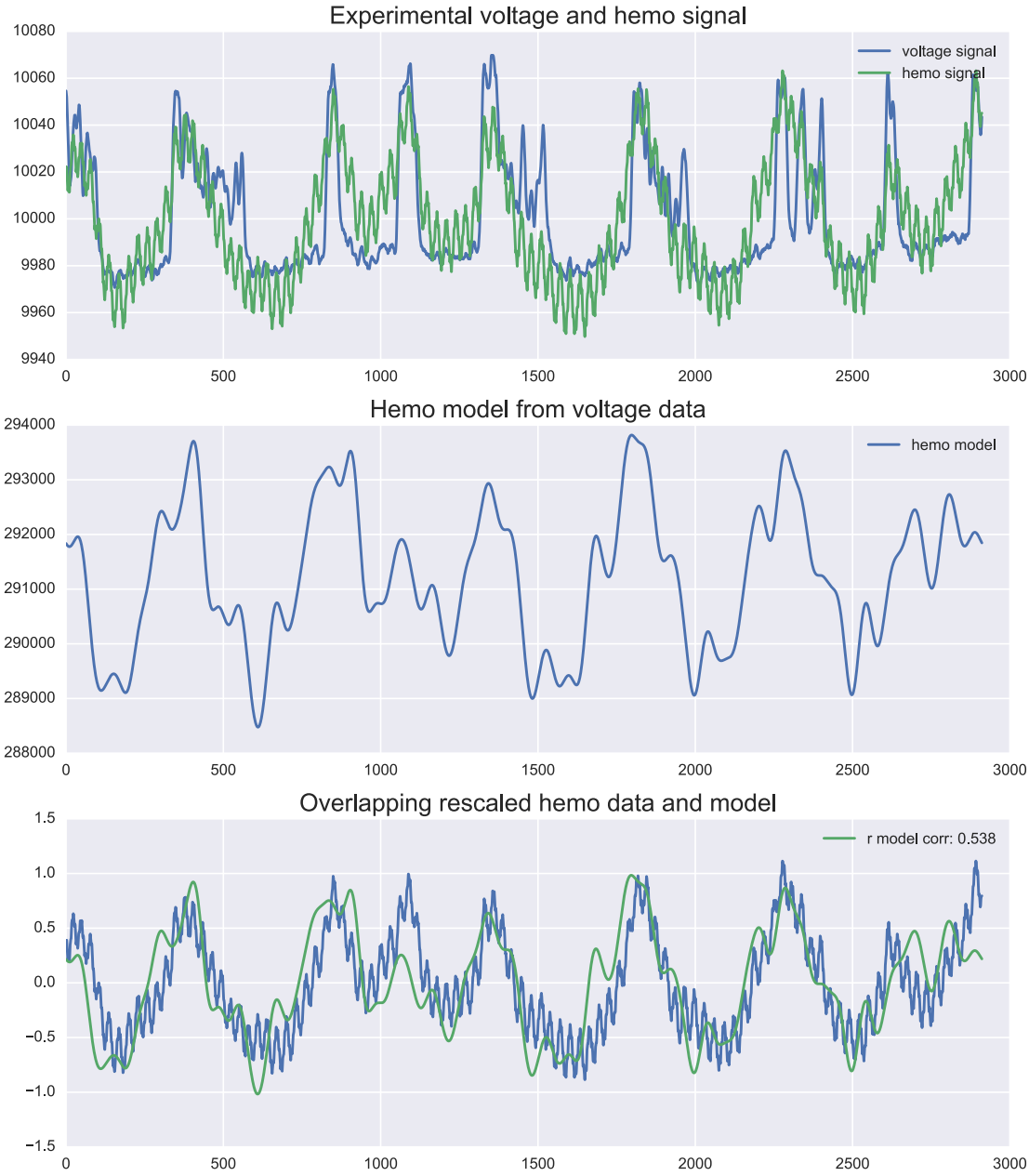



```
In [22]: plotRModel(dataB, mouse = 'B', exp = 2, repeat = 4,alpha = getMeanAlphaFiltered([dataA[2]],0,1
         plotHModel(dataB, mouse = 'B', exp = 2, repeat = 4,alpha = getMeanAlphaFiltered([dataA[2]],0,1
```

Voltage Mouse B - Exp 4 - Rep 5

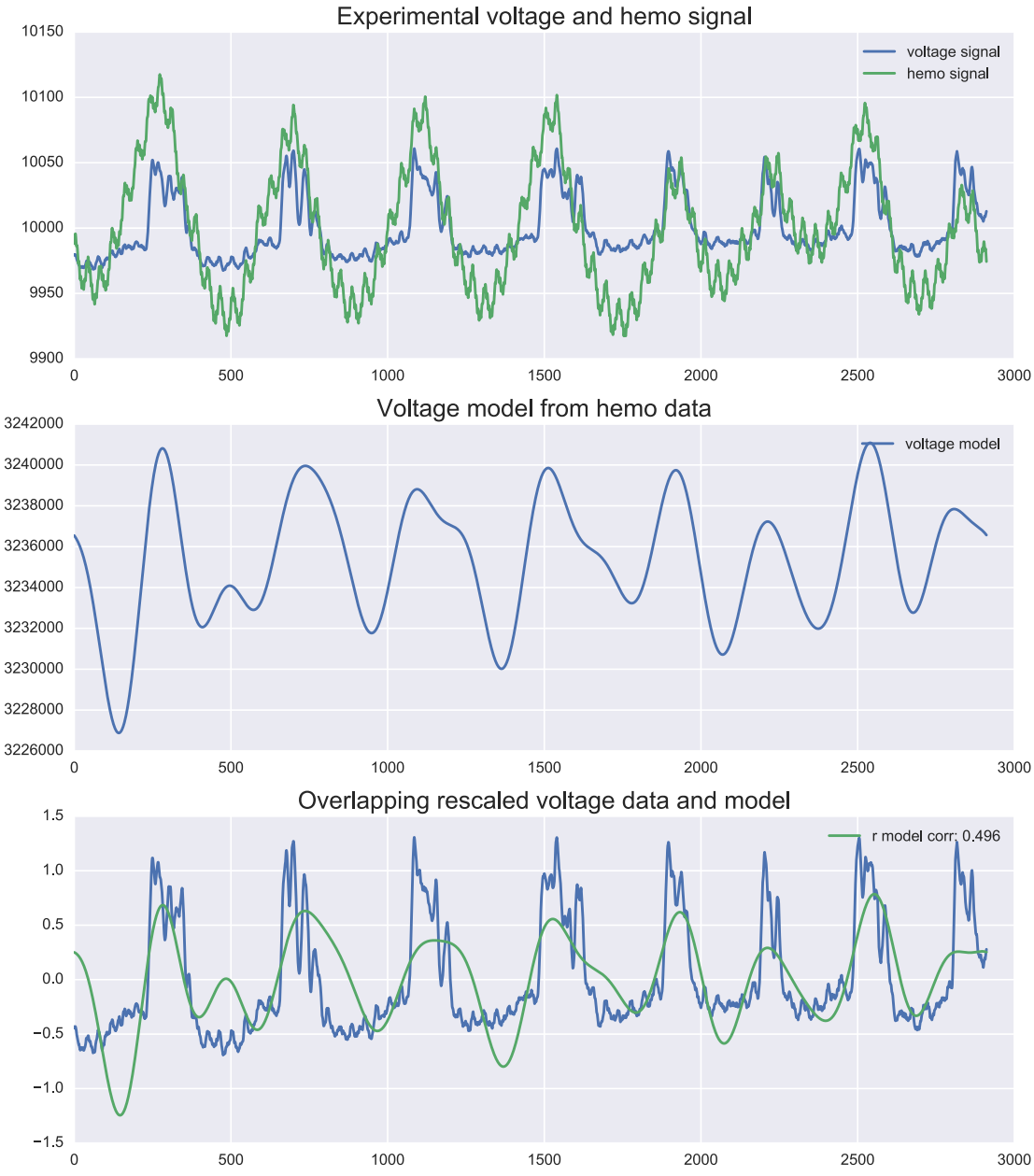


Hemo. Mouse B - Exp 4 - Rep 5




```
In [23]: plotRModel(dataB, mouse = 'B', exp = 0, repeat = 2,alpha = getMeanAlphaFiltered([dataA[2]],0,1
         plotHModel(dataB, mouse = 'B', exp = 0, repeat = 2,alpha = getMeanAlphaFiltered([dataA[2]],0,1
```

Voltage Mouse B - Exp 2 - Rep 3



Hemo. Mouse B - Exp 2 - Rep 3