What I’ve been up to...

- Not much time to go into depth
- But, I’ll start at the beginning
- Arduino: founded in 2008
- Microcontrollers exploded from there
  - Unique features of the Arduino:
    - Cheap!!
    - Open source!!
    - Simple programming language and environment (written in language similar to C++, and in fact can utilize C++ and C libraries

Other advantages

- Only a single process is run at a time: no overhead of an operating system, etc.
- Easy to build and add shields, attach to other boards, etc.
- Lend themselves VERY easy to experimental control, and are very accurate as noted and tested previously noted in “Arduino: A low-cost multipurpose lab equipment” (Alessandro D’Ausilio, 2012).

Teensy microcontrollers

- Basically, the same thing, with a few unique features
- Has a library built on top of the Arduino library (i.e. can use Arduino functions, plus many extras!)
- We utilize the 3.2 in particular
- Similar number of channels, but capable of true analog output (instead of PWM)
- Faster processor (potentially more accuracy): 72 vs 16 MHz
- All pins have interrupt capability (we don’t utilize this here)
- Special, added libraries on top of the standard Arduino libraries

Teensy libraries

- Key functions: IntervalTimer → generates interrupts at microsecond time scale; can utilize 4 at a time
- Audio library: 44.1 kHz library for playing and synthesizing tones – super easy to use, and can be used concomitantly with other interval timers
- Plug-and-play add-ons: e.g. prop-shield serves as an easy amplifier: just solder on, and you are good to go
- Like Arduino, capable of interacting with serial peripheral interfaces, like the ADNS-9800

ADNS-9800

- Originally a gaming sensor
- Board created recently by Jack Enterprises (https://www.tindie.com/products/jkicklighter/adns-9800-laser-motion-sensor/)
- 12000 FPS frame rate, up to 8200 counts per inch (compare to ~ 125 fps max, maybe 1000, and far fewer counts per inch)
- 16 bit storage: don’t have to poll as often, so the sensors won’t saturate

ADNS-9800

- Simple SPI interface
- *Our* Mark expanded the included code into a library, so it is easy to start sensors, clear, and read from motion burst registers and others using his code instead of diving into the nitty gritty (back in 2014...)
- Downsides: wiring the sensors kind of sucks

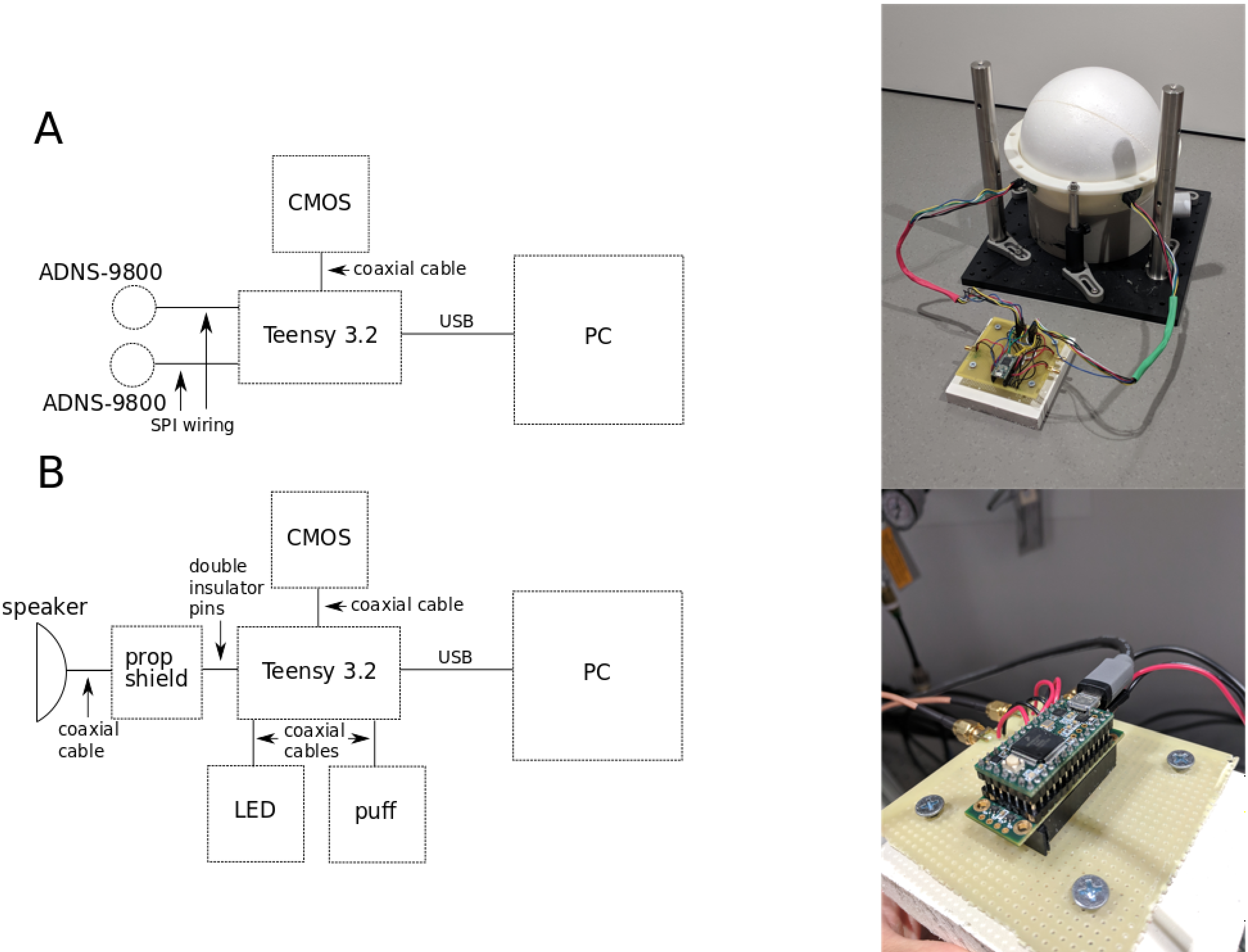

A
CMOS
coaxial cable
ADNS-9800
Teensy 3.2
USB
PC
ADNS-9800
SPI wiring
B
CMOS
double
insulator
pins
coaxial cable
speaker
prop
shield
Teensy 3.2
USB
PC
coaxial
cable
coaxial
cables
LED
puff

Figure 2 (audio and motor schematic)

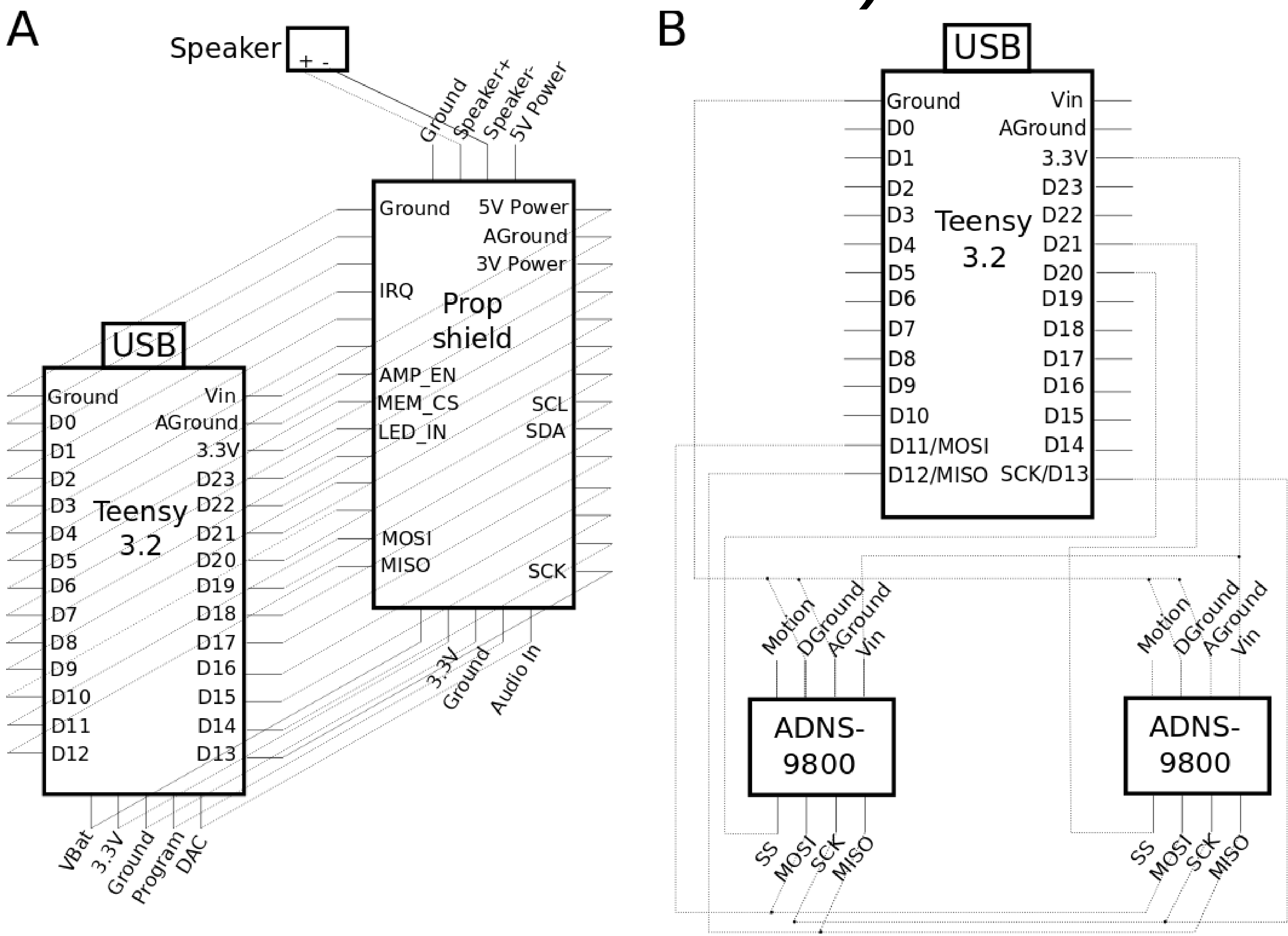
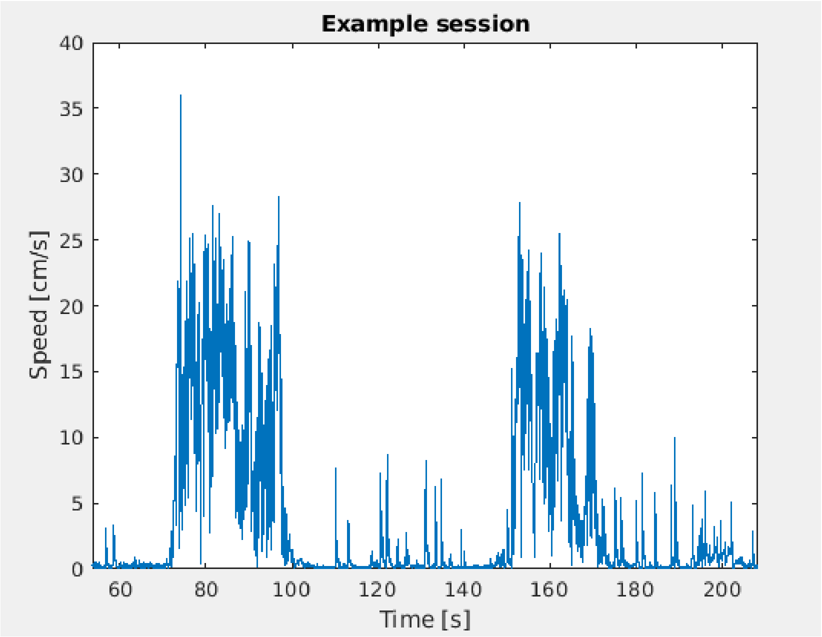
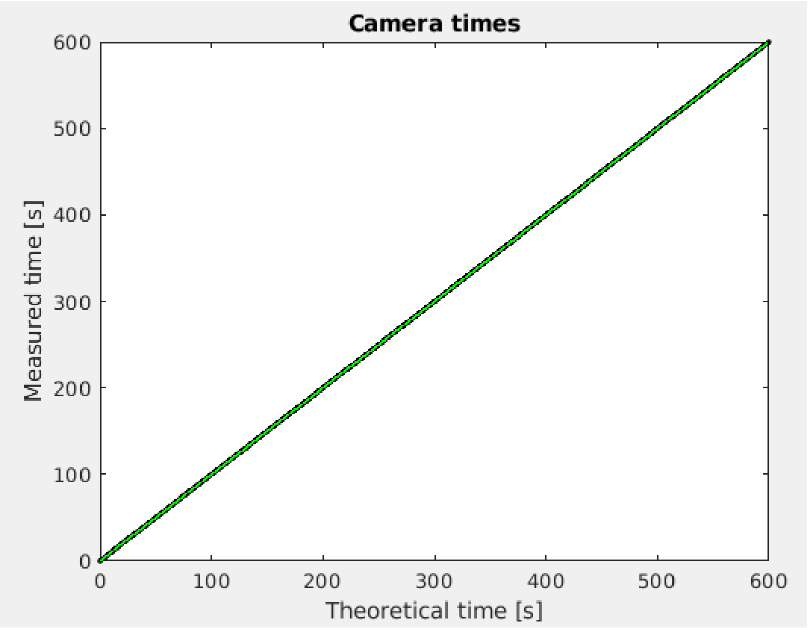


Figure 3





Data collected from the device is temporally VERY accurate, only slightly biased

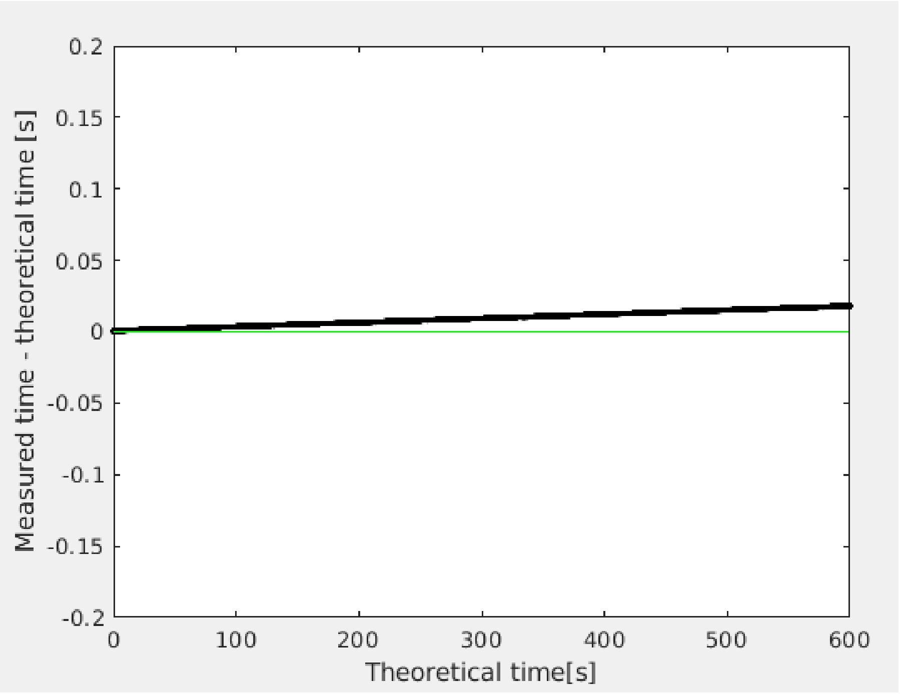
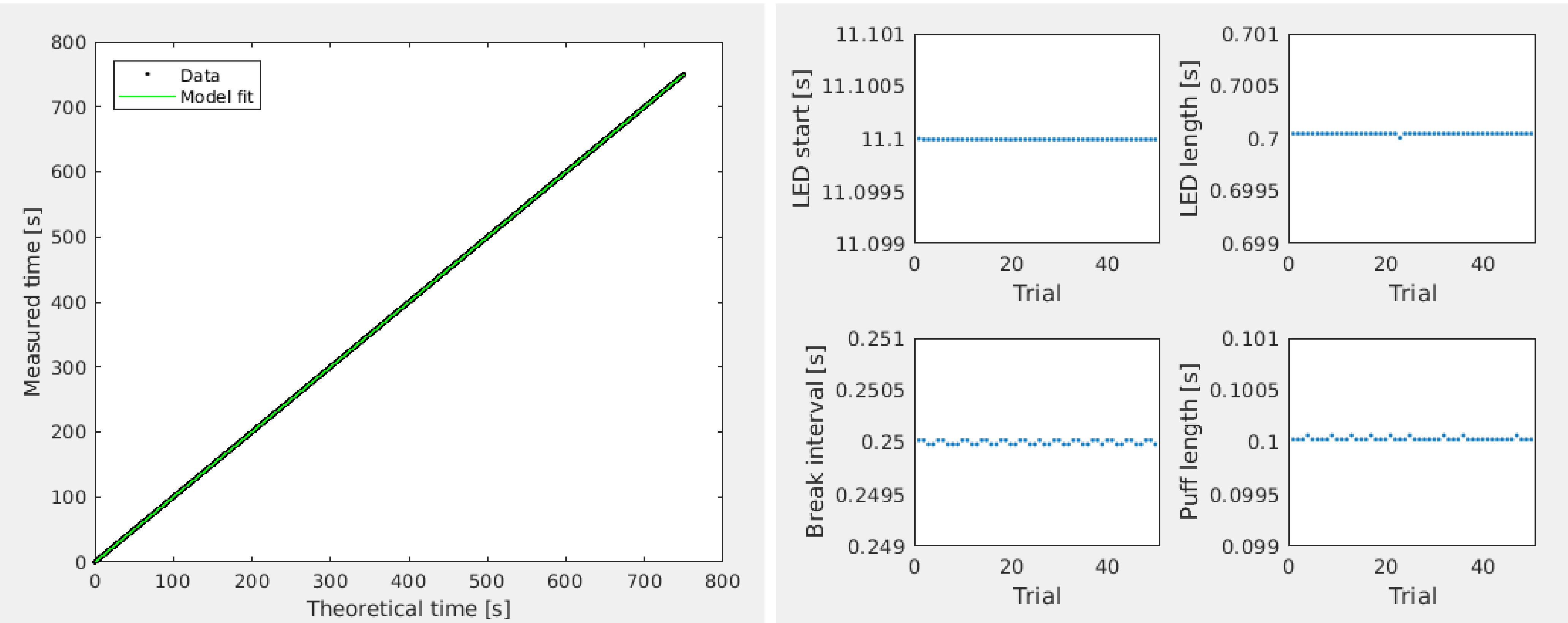


Figure 4



Tone puff: also very accurate

Conclusion

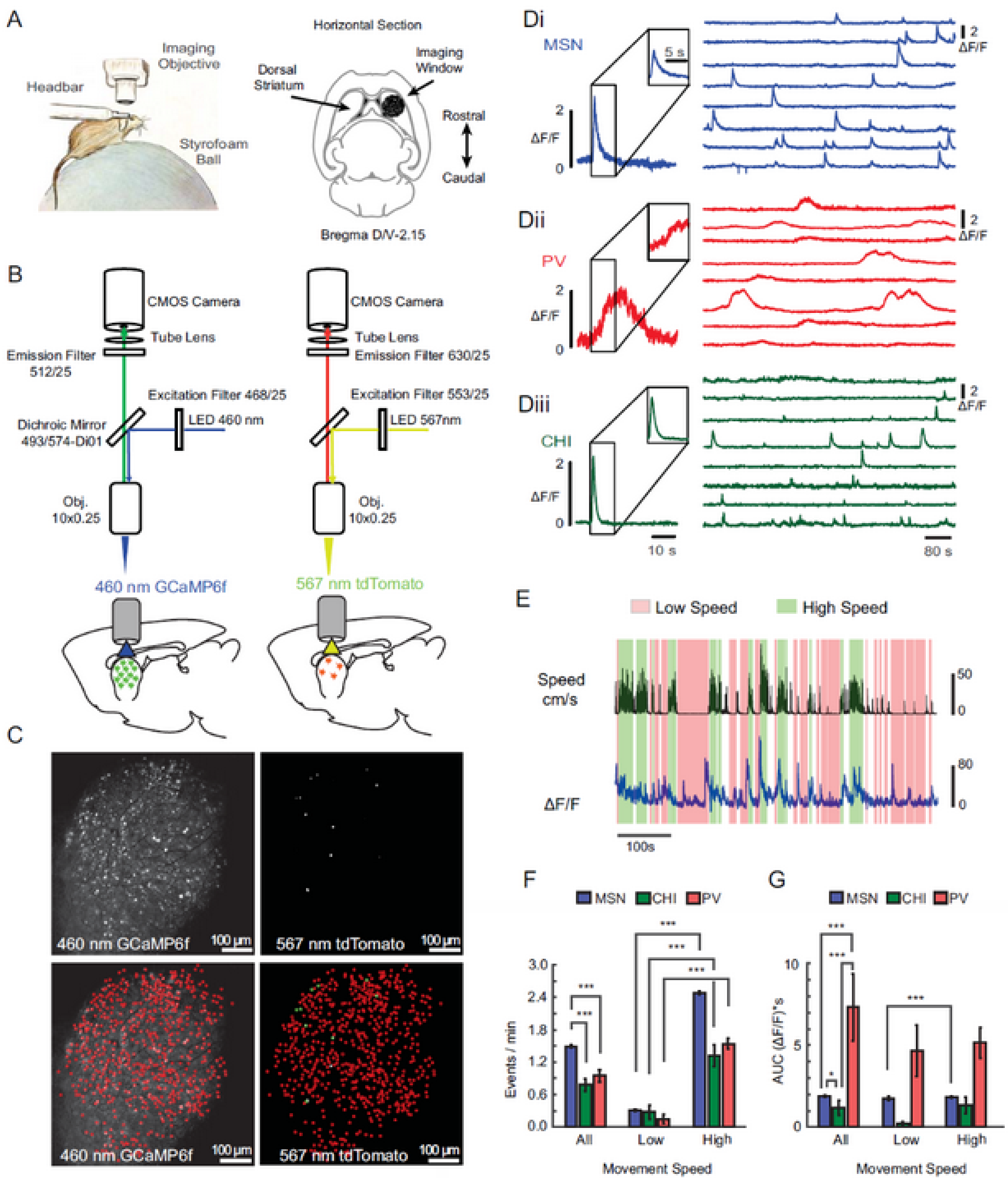
- Use these!
- Excellent temporal resolution; not as high-level as MATLAB: less jitter/competing processes
- Easy to set up
- Should be easy to integrate into ViRMeN (sp?), if anyone wants a project?

Revisions to the original manuscript

1 The words of the Teacher,[a] son of David, king in Jerusalem.

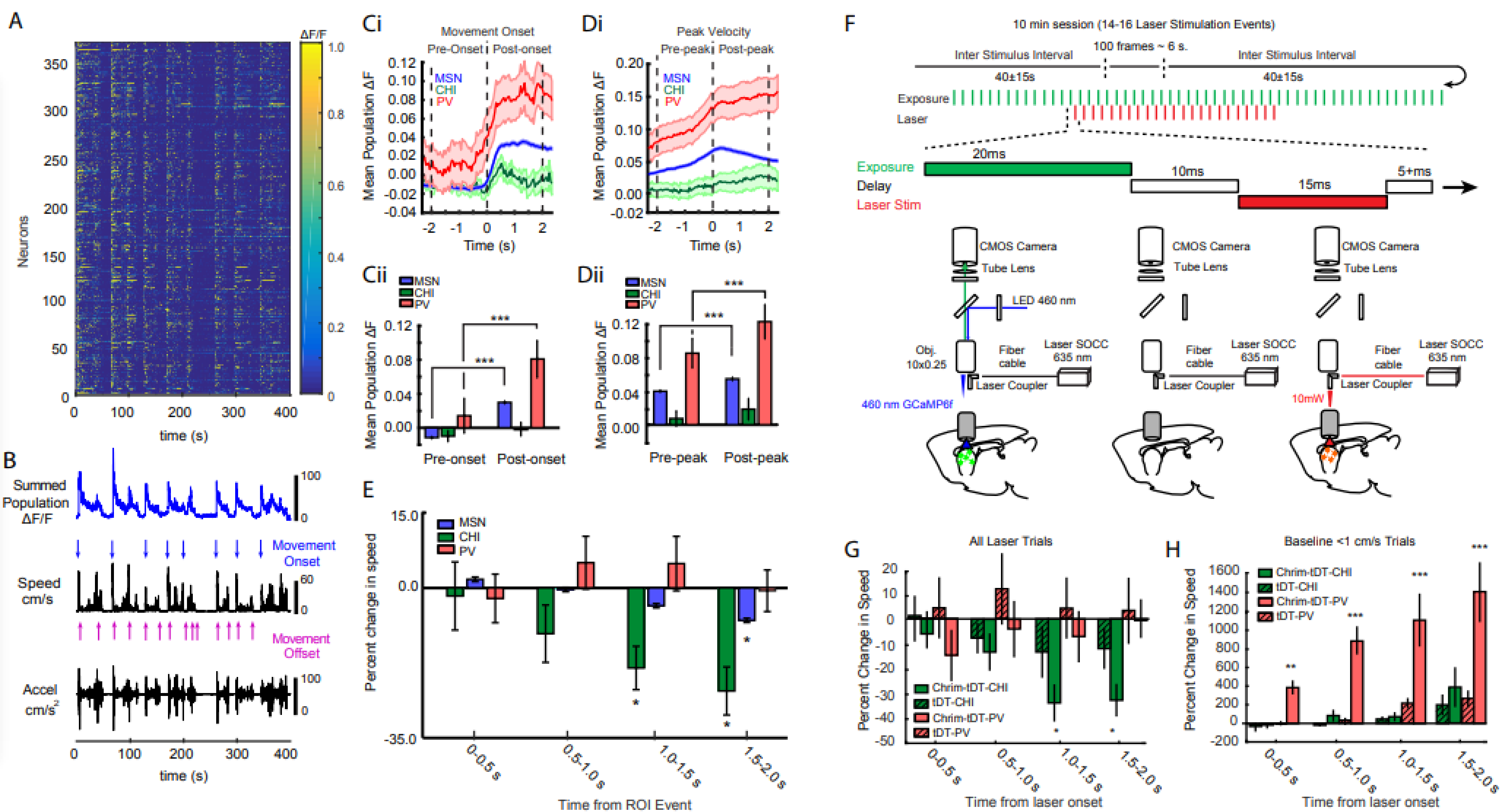
2 “Absolute futility,” says the Teacher.
“Absolute futility. Everything is futile.”
3 What does a person gain for all his efforts
that he labors at under the sun?
4 A generation goes and a generation comes,
but the earth remains forever.
5 The sun rises and the sun sets;
panting, it returns to the place
where it rises.
6 Gusting to the south,
turning to the north,
turning, turning, goes the wind,
and the wind returns in its cycles.
7 All the streams flow to the sea,
yet the sea is never full;
to the place where the streams flow,
there they flow again.
8 All things[b] are wearisome,
more than anyone can say.
The eye is not satisfied by seeing
or the ear filled with hearing.

Figure1



A
Imaging Objective
Headbar
Styrofoam Ball
Horizontal Section
Dorsal Striatum
Imaging Window
Rostral
Caudal
Bregma D/V-2.15
B
CMOS Camera
Tube Lens
Emission Filter 512/25
Excitation Filter 468/25
LED 460 nm
Dichroic Mirror 493/574-Di01
Obj. 10x0.25
460 nm GCaMP6f
CMOS Camera
Tube Lens
Emission Filter 630/25
Excitation Filter 553/25
LED 567nm
Obj. 10x0.25
567 nm tdTomato
C
460 nm GCaMP6f
100 µm
567 nm tdTomato
100 µm
460 nm GCaMP6f
100 µm
567 nm tdTomato
100 µm
Di
MSN
5 s
ΔF/F
2 ΔF/F
Dii
PV
ΔF/F
2 ΔF/F
Diii
CHI
ΔF/F
2 ΔF/F
10 s
80 s
E
Low Speed
High Speed
Speed cm/s
50
0
ΔF/F
80
0
100s
F
MSN CHI PV
Events / min
3.0
2.4
1.8
1.2
0.6
0.0
All
Low
High
Movement Speed
G
MSN CHI PV
AUC (ΔF/F)*s
10
5
0
All
Low
High
Movement Speed

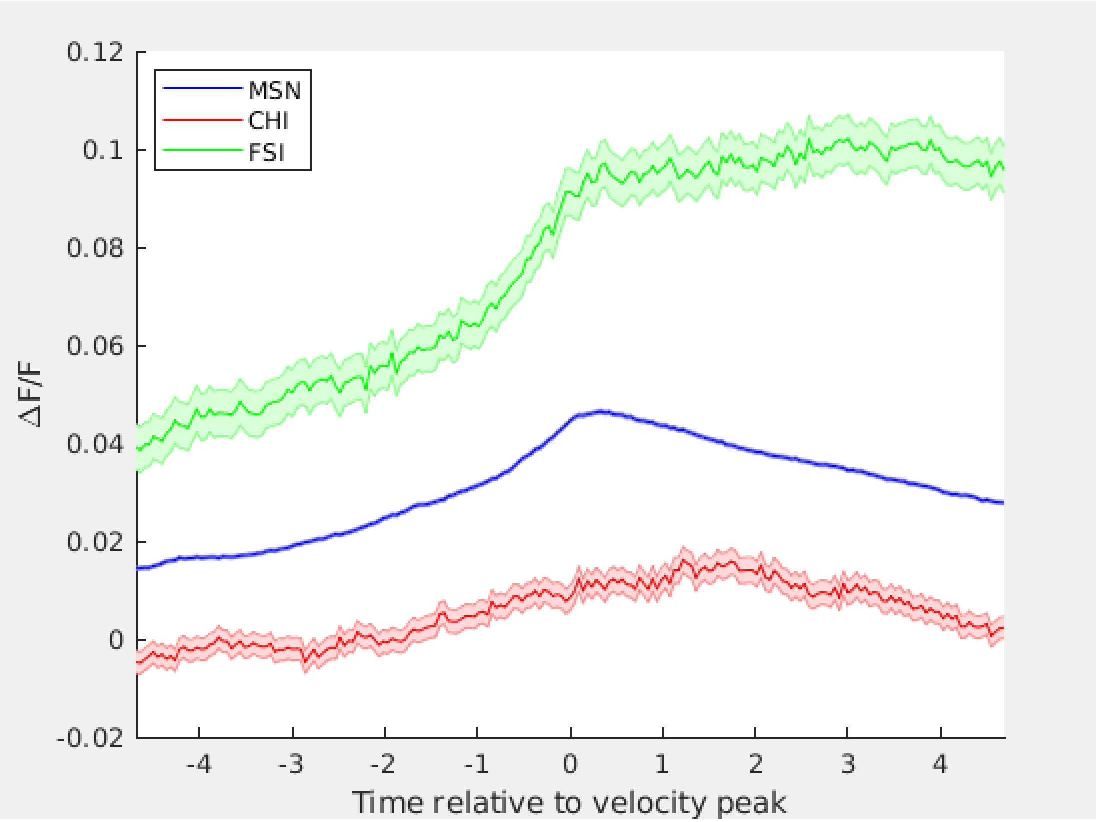
Figure 2



Issues: do PV cells increase their firing before MSNs?

Do CHI interneurons peak later than MSNs?

Also, how do we know that percent change in speed is different for three cell types?




```
Linear mixed-effects model fit by ML

Model information:
    Number of observations          6108216
    Fixed effects coefficients           18
    Random effects coefficients       11672
    Covariance parameters                 4

Formula:
    Fluor ~ 1 + TimeBin*CellType + (1 | EventID) + (1 | ROIID) + (1 | MouseID)

      ANOVA marginal tests: DFMethod = 'Residual'

      Term                        FStat     DF1     DF2             pValue
      '(Intercept)'               19.417     1      6.1082e+06      1.0507e-05
      'TimeBin'                   437.45     5      6.1082e+06               0
      'CellType'                  12.923     2      6.1082e+06      2.4413e-06
      'TimeBin:CellType'          9.5222    10      6.1082e+06      4.9072e-16
```

```
Formula:
    Fluor ~ 1 + TimeBin + (1 | ROIID) + (1 | MouseID) + (1 | EventID)
```

FSI

```
Benjamini-Hochberg corrected p-values (across all cell types)
    TimeBin 2 vs TimeBin1: p=2.780985e-03
    TimeBin 3 vs TimeBin1: p=6.805390e-11
    TimeBin 4 vs TimeBin1: p=4.066278e-12
    TimeBin 5 vs TimeBin1: p=8.886293e-14
    TimeBin 6 vs TimeBin1: p=1.744670e-14
```

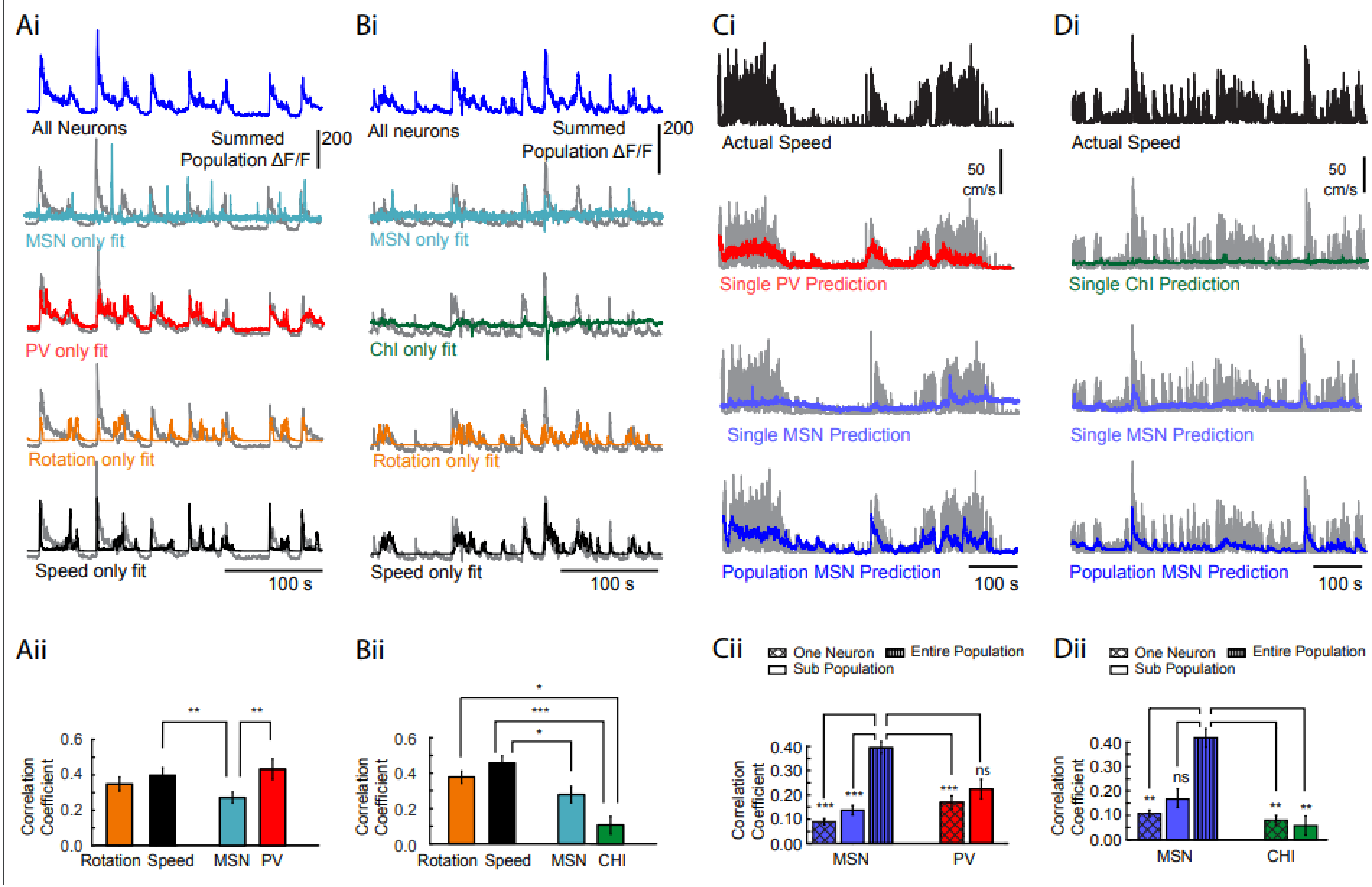
MSN

```
Benjamini-Hochberg corrected p-values (across all cell types)
    TimeBin 2 vs TimeBin1: p=3.340711e-103
    TimeBin 3 vs TimeBin1: p=0
    TimeBin 4 vs TimeBin1: p=3.822626e-217
    TimeBin 5 vs TimeBin1: p=3.794100e-95
    TimeBin 6 vs TimeBin1: p=7.300842e-29
```

CHI

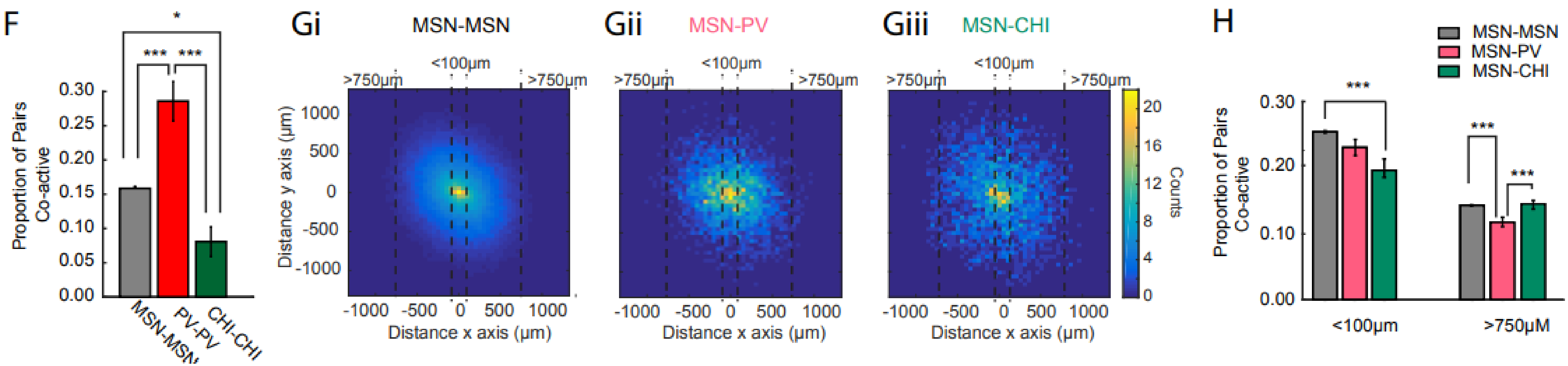
```
Benjamini-Hochberg corrected p-values (across all cell types)
    TimeBin 2 vs TimeBin1: p=6.500318e-02
    TimeBin 3 vs TimeBin1: p=4.277905e-03
    TimeBin 4 vs TimeBin1: p=8.851190e-05
    TimeBin 5 vs TimeBin1: p=4.296597e-03
    TimeBin 6 vs TimeBin1: p=6.500318e-02
```

Figure 4

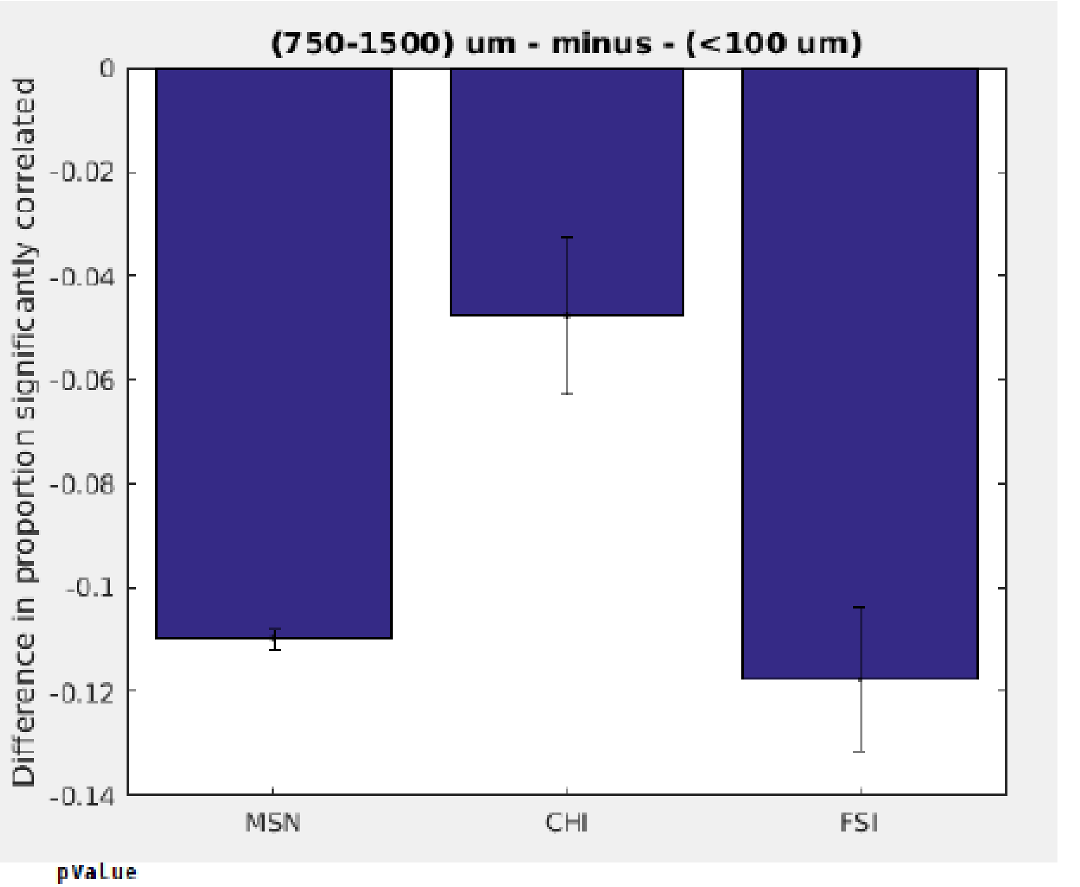


Criticism: why not combine everything into a single model and thereby make inferences?

```
Linear mixed-effects model fit by ML

Model information:
    Number of observations          46899980
    Fixed effects coefficients             6
    Random effects coefficients        15826
    Covariance parameters                  7

Formula:
    speeds ~ 1 + speedlag1 + speedlag2 + traces + traces:roi_type + (1 + traces | mouseno) + (1 + traces | roi_no)

Model fit statistics:
    AIC           BIC           LogLikelihood    Deviance
    9.0377e+07    9.0377e+07    -4.5189e+07      9.0377e+07

Fixed effects coefficients (95% CIs):
    Name                             Estimate      SE            tStat       DF            pValue        Lower          Upper
    '(Intercept)'                    5.9493e-06    9.3792e-05    0.063431    4.69e+07         0.94942    -0.00017788    0.00018978
    'speedlag1'                         0.61673    0.00014355      4296.3    4.69e+07               0        0.61645       0.61701
    'speedlag2'                         0.18516    0.00014357      1289.7    4.69e+07               0        0.18488       0.18544
    'traces'                           0.016864     0.0014891      11.325    4.69e+07      9.9639e-30       0.013946      0.019783
    'traces:roi_type_fsi'              0.023676     0.0032045      7.3884    4.69e+07      1.4858e-13       0.017396      0.029957
    'traces:roi_type_chi'             -0.019356      0.003979     -4.8645    4.69e+07      1.1477e-06      -0.027154     -0.011557

Random effects covariance parameters (95% CIs):
Group: mouseno (28 Levels)
    Name1                  Name2                  Type           Estimate        Lower      Upper
    '(Intercept)'          '(Intercept)'          'std'          7.8401e-05      NaN        NaN
    'traces'               '(Intercept)'          'corr'           -0.99994      NaN        NaN
    'traces'               'traces'               'std'           0.0075964      NaN        NaN

Group: roi_no (7885 Levels)
    Name1                  Name2                  Type           Estimate        Lower         Upper
    '(Intercept)'          '(Intercept)'          'std'          4.7489e-09           NaN           NaN
    'traces'               '(Intercept)'          'corr'         -0.0015694           NaN           NaN
    'traces'               'traces'               'std'            0.026926      0.026469      0.027391

Group: Error
    Name               Estimate    Lower      Upper
    'Res Std'          0.63404     0.63391    0.63417
```



Reviewer comment: argue that the change in proportion of pairs coactive vs distance is different for all three cell types, but demonstrate no statistics



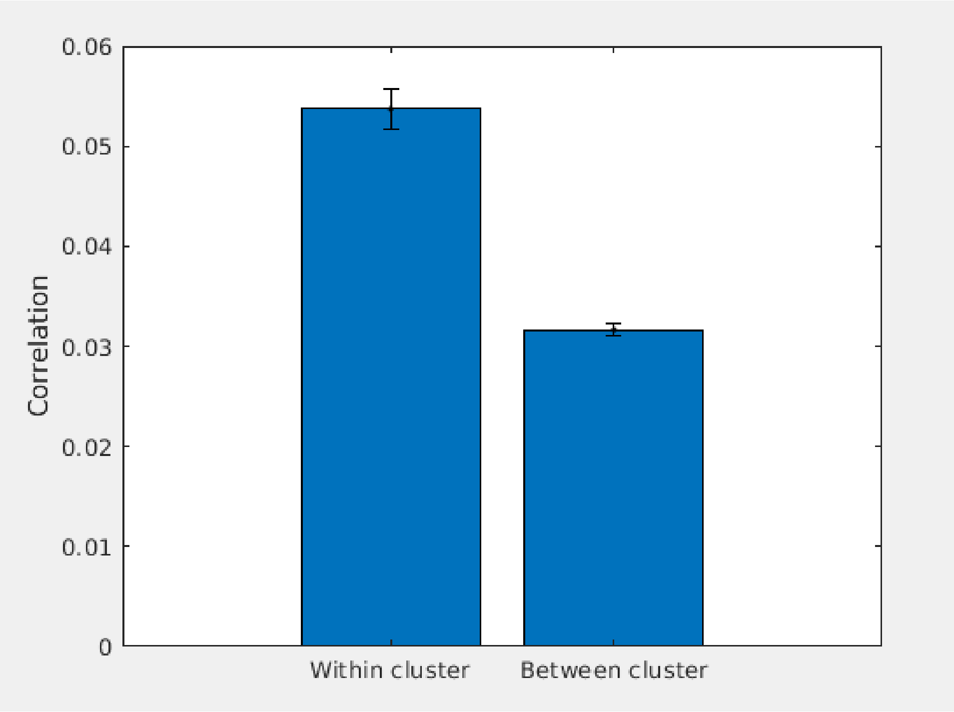

```
Generalized Linear regression model:
    is_sig ~ 1 + binids*genotype
    Distribution = Binomial

Estimated Coefficients:
                                   Estimate        SE         tStat        pValue
                                  __________    _________    ________    __________

    (Intercept)                      0.25285    0.0017291      146.24             0
    binids_3                        -0.10971    0.0018947     -57.904             0
    genotype_msn-chi               -0.058805     0.013754     -4.2755    1.9073e-05
    genotype_msn-fsi               -0.023635     0.012066     -1.9589       0.05013
    binids_3:genotype_msn-chi        0.06204     0.015229      4.0739    4.6242e-05
    binids_3:genotype_msn-fsi     -0.0078585     0.013925    -0.56433       0.57253


274589 observations, 274583 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 3.95e+03, p-value = 0
```

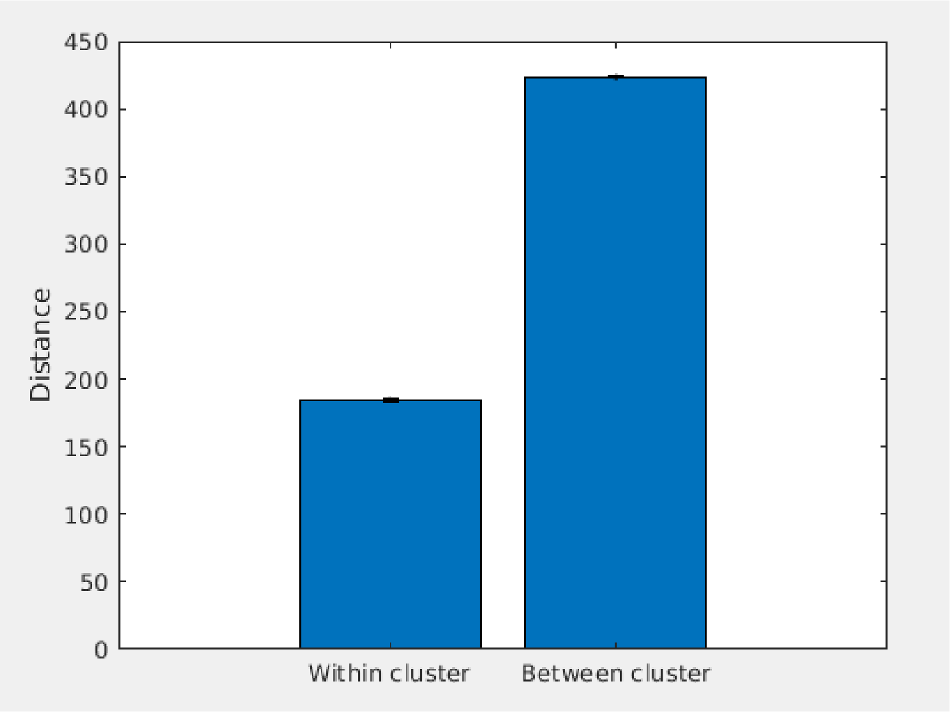
```
Coefficients for interaction term corresponding to the following correlation types
binids_3:genotype_msn-msn-vs-binids_3:genotype_msn-chi: t(274583)=4.073853, p=1.387248e-04
binids_3:genotype_msn-msn-vs-binids_3:genotype_msn-fsi: t(274583)=-0.564331, p=5.725287e-01
binids_3:genotype_msn-chi-vs-binids_3:genotype_msn-fsi: t(274583)=-3.416175, p=9.526117e-04
---------------------------
```

Now, beginning a “network” analysis

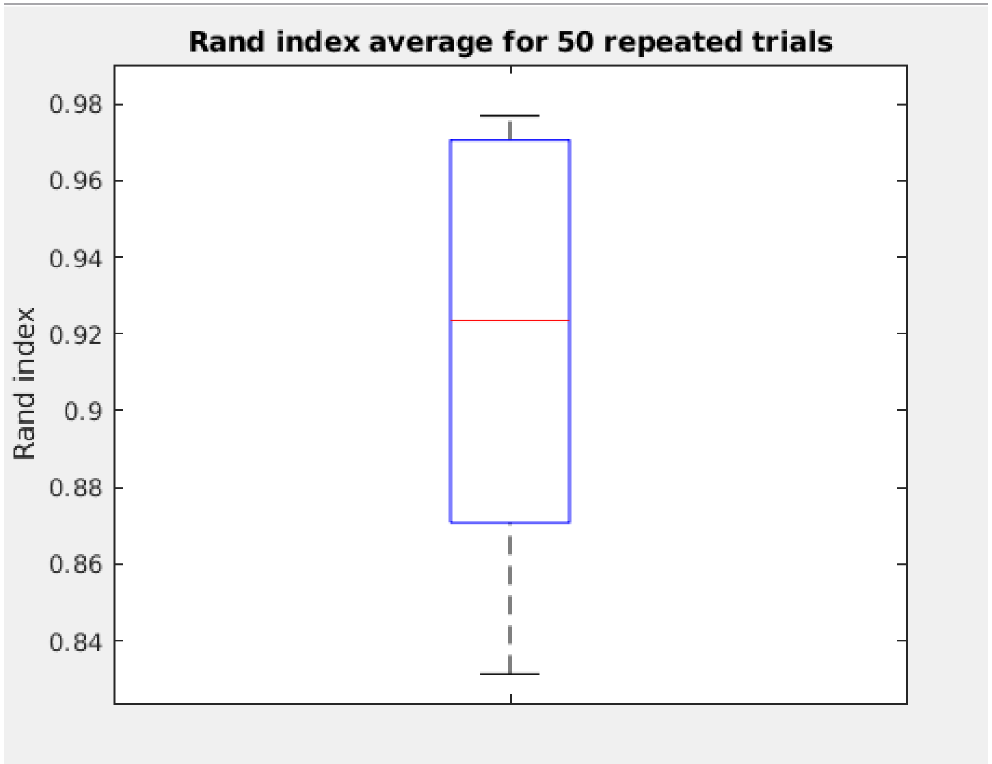
- Mentioned over the summer, several papers have reported that neurons form clusters that are spatially biased
- What do we see when we use the same methodology? (meta k-means)
  - 1st, randomly select chunk of data
  - Run k-means 100 times with “correlation” as distance metric
  - Randomly select different chunk of data
  - Make histogram: # of times that each pair together
  - Choose cutoff value that maximizes # of clusters

Summary statistics

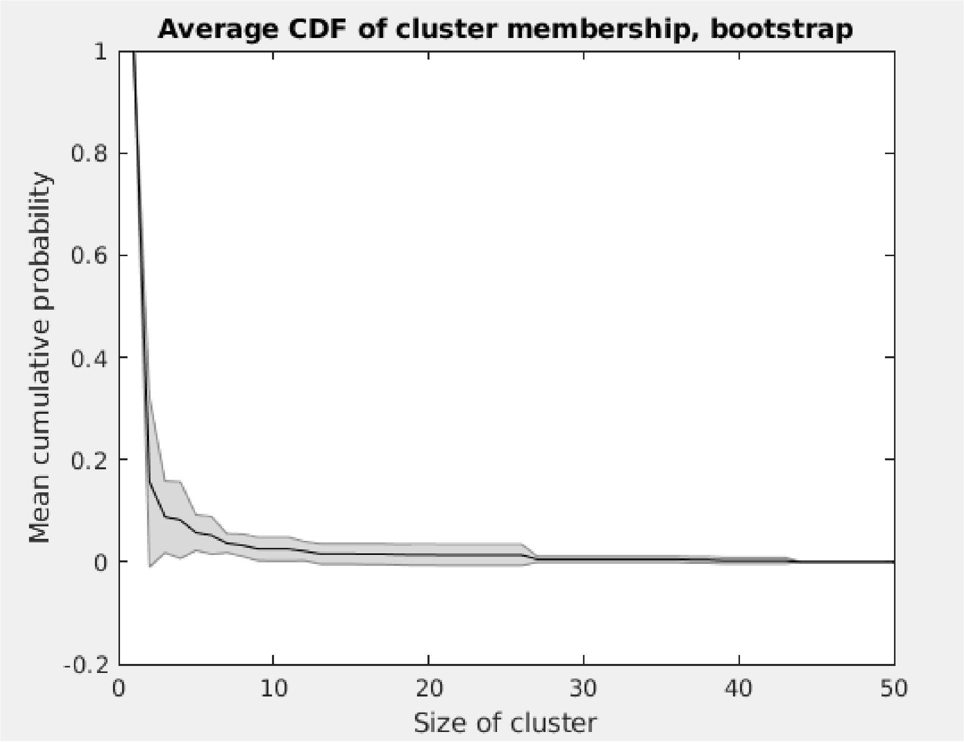


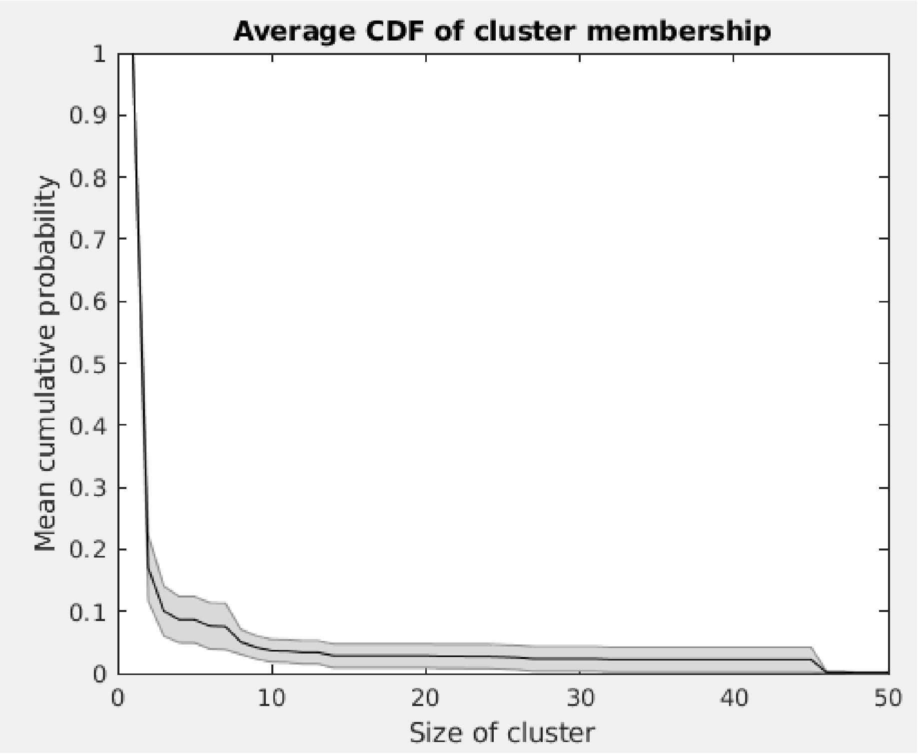


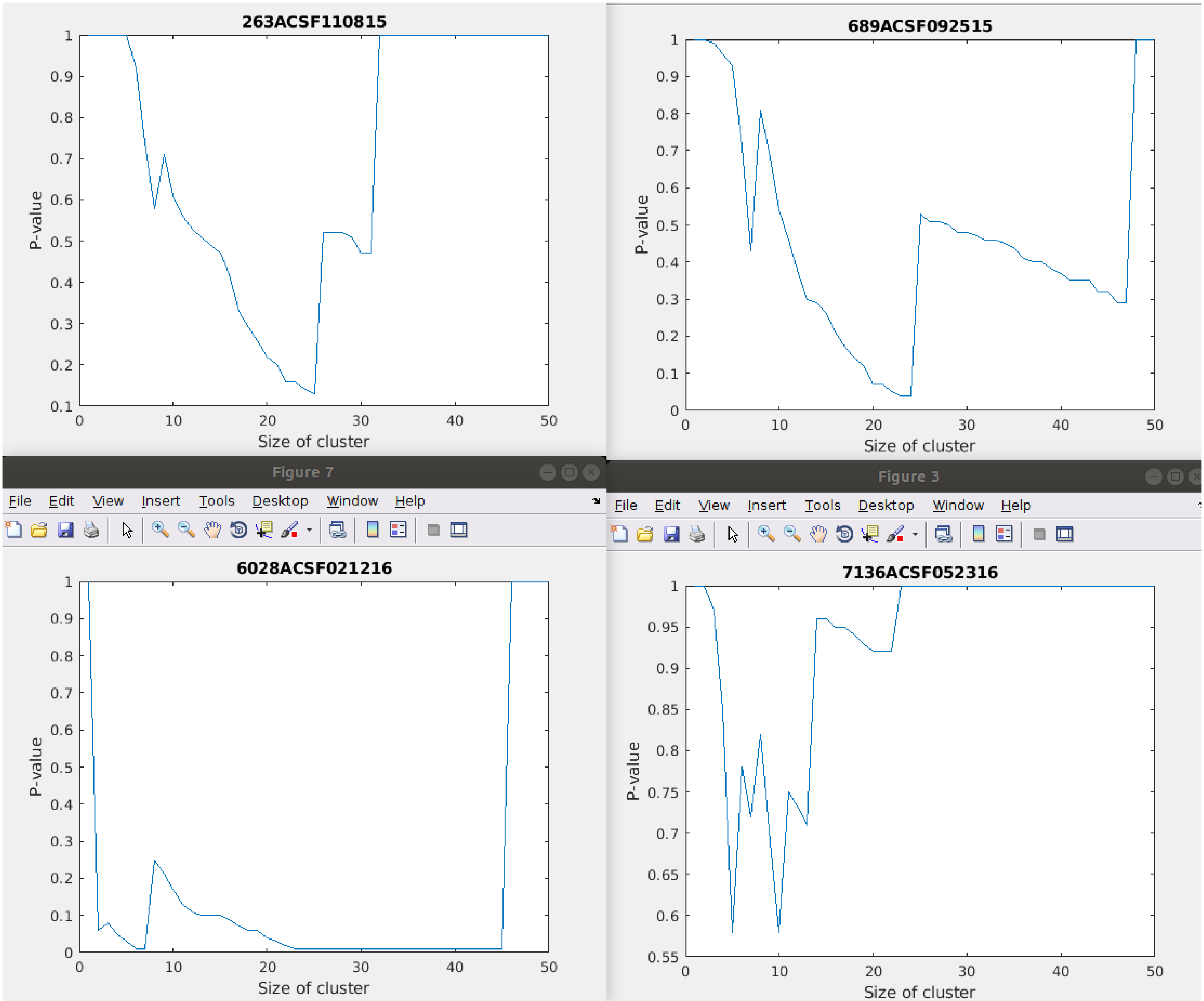
Make sure we are converging on same clusters



Summary statistics







263ACSF110815
P-value
Size of cluster
689ACSF092515
P-value
Size of cluster
Figure 7
File Edit View Insert Tools Desktop Window Help
6028ACSF021216
P-value
Size of cluster
Figure 3
File Edit View Insert Tools Desktop Window Help
7136ACSF052316
P-value
Size of cluster

Next steps

- Keep going down this path? Then, see whether or not clustering