OFFICIAL Cell Types Projects FINAL.Cell Type 5

March 13, 2020

1 Exploring electrophysiology between CA1 and CA3 regions

1.1 Final Project

1.1.1 Cell Type 5

2 Abstract

CA1 and CA3 neurons are a vital part of the hippocampal circuitry, to which various studies have argued that the subregions differ in terms of the electrophysiology. We hypothesized that CA1 and CA3 had a difference in their general firing rates, repolarization, and waveform amplitude. Using both LISC, Neuroelectro, and Allen brain atlas, we were able to extract data that allowed us to observe study correlations and electrophysiological properties from hippocampal CA1 and CA3 cells including the variables we meant to examine. Ater visualizing our data, we concluded that CA1 and CA3 regiond were significantly difference from each other in terms of their firing rate ($p<0.017$), waveform repolarization slope ($p<0.017$), and waveform amplitude ($p<0.017$) with all of them greatly skewed (all; $p<0.05$), and CA3 (repolarization=0.46, firing rate=6.11, amplitude=143.99) having higher median values in all three variables comapred to CA1 (repolarization=0.40, firing rate=4.70, amplitude=131.26).

3 Research Question

Could the pyramidal neurons in subregions CA1 and CA3 of the Hippocampus be better distinguished as their own unique regions by comparing single-cell electrophysiological and Local Field Potential (LFP) data via firing rates, repolarization, and waveform amplitudes?

4 Background

The hippocampus is a medial temporal lobe structure that is critical for learning and memory [1], fear processing, spatial navigation and other cognitive functions [2]. It is divided into the dorsal hippocampus (DHC) and the ventral hippocampus (VHC) with each area consisting of the dentate gyrus (DG) and the Cornu Ammonis (CA) region. Within those regions are several sub-regions including CA1, CA2, CA3, and CA4. [3] It has become increasingly apparent of the anatomical and functional differences between the dorsal and ventral hippocampus. For instance, the dorsal component is associated with spatial navigation while the ventral component is associated with emotional responses [3]. The reason for that is that the dorsal section receives input from the sensory cortices through the medial entorhinal cortex while the ventral section is connected to the amygdala, prefrontal cortex, and the hypothalamus [2]. This would imply that the CA sub-regions within these hippocampal areas would be different as well.

There is evidence that mark specific distinctions regarding the electrophysiological and morphological properties between CA1 and CA3. For example, the CA1 region is characterized by homogenous pyramidal neurons with apical dendrites and apical and basilar dendritic trees similar in length while the CA3 region contains heterogeneous pyramidal neurons with different dendritic lengths [3]. This is particularly important when it comes to long-term potentiation (LTP). Neurons exhibiting firing patterns that consist of bursts of spikes can more effectively induce LTP than single spikes [4]. In Elburg & Ooyen's study, it was found that as the length of the dendritic tree becomes shorter, the degree of bursts decreases as well [4]. This suggests that shorter dendritic trees decrease the likelihood of achieving LTP. There's also recognition that CA1 pyramidal cells and CA3 pyramidal cells are responsive with different connections between other regions in the brain, thus having their own unique functions to different stimuli [3]. In Henze & Patrick Card's study, CA3 pyramidal cells produce large amplitudes from miniture excitatory postsynaptic currents due to the involvment of mossy fibers, which are only in the boundaries of CA3 and not CA1 [6]. While in Storm's study, during repetative firing in CA1, it was observed that CA1 pyramidal cells produced fast after hyperpolarizations due to being Ca2+ dependent [7]. But the occurance of these fast AHP's are also present for all parts in the Hippocampus, specifically when its due to the same current used for producing spike repolarizations[7]. Due to CA3 neuons having different dendritic characteristics than CA1 neurons, then under the same current stimulus, would their AP's depolarized amplitudes and spike repolarizations and be different due to other influences surrounding those two regions that uniquely discriminates them to begin with? If the pyramidal cells have different structural and electrophysiological features, then should there be a difference in firing rates between CA1 and CA3? If so, there may be other variables that could contribute to CA1 and CA3 alone to produce these differneces.

This motivates us to validate these article's findings and observe multiple characteristics of generated action potentials: firing rates, repolarization, and waveform amplitude, between the two subregion that will allow us to determine whether or not there is a true difference regarding the electrophysiology of CA pyramidal cells. Although there are other types of understanding the network activity, within these regions can provide more insight on hippocampal function and can open more doors in the field of neuroscience. For this reason, we seek to explore differences between CA1 and CA3 subregions by examining and comparing single-cell electrophysiological and Local Field Potential (LFP) data.

4.0.1 References:

- [1] R. A. J. van, & Ooyen, A. van. (2013, May 13). "Impact of Dendritic Size and Dendritic Topology on Burst Firing in Pyramidal Cells." Retrieved from https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000781Milior
- [2] A. B., Aguilar-Salgado, Y., Reyes-Hernández, D. O., & Flores, G. (2012, December 26). "Dexamethasone induces different morphological changes in the dorsal and ventral hippocampus of rats." Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0891061812000804

- [3] Milior, G., Castro, M. A. D., Sciarria, L. P., Garofalo, S., Branchi, I., Ragozzino, D., … Maggi, L. (2016, December 6). "Electrophysiological Properties of CA1 Pyramidal Neurons along the Longitudinal Axis of the Mouse Hippocampus." Retrieved from https://www.nature.com/articles/srep38242
- [4] Dougherty, K. A., Islam, T., & Johnston, D. (2012, November 15). "Intrinsic excitability of CA1 pyramidal neurones from the rat dorsal and ventral hippocampus." Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3528986/Elburg
- [5] Mercer, A., Trigg, H. L., & Thomson, A. M. (2007, July 4). "Characterization of neurons in the CA2 subfield of the adult rat hippocampus." Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6794598/
- [6] Darrell , Henze A., et al. (1997, March 1). "Large Amplitude Miniature Excitatory Postsynaptic Currents in Hippocampal CA3 Pyramidal Neurons Are of Mossy Fiber Origin." Journal of Neurophysiology, vol. 77, no. 3, pp. 1075–1086. Retrieved from https://journals.physiology.org/doi/full/10.1152/jn.1997.77.3.1075.
- [7] Storm, J F. "Action potential repolarization and a fast after-hyperpolarization in rat hippocampal pyramidal cells." The Journal of physiology vol. 385 (1987): 733-59. doi:10.1113/jphysiol.1987.sp016517

5 Hypothesis

We hypothesize that CA1 pyramidal neurons will have different firing rates, repolarizations, and waveform amplitudes than CA3 pyramidal neurons. Since there are clear anatomical and functional differences between these two sub regions, such as their dendritic lengths, there must also be a difference in their electrophysiological properties.

6 Data Analysis

6.1 Setup

Installs various packages allowing the use of statistics, PCA, plots, and Allensdk.

[1]:

```
import os
import numpy as np
import pandas as pd
from urllib.request import urlopen
from pandas import read_excel

from allensdk.brain_observatory.ecephys.ecephys_project_cache import␣
 ↪EcephysProjectCache

import matplotlib as mpl
import matplotlib.pyplot as plt

from scipy import stats
import seaborn as sns
import scipy as sp

from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
```

Installs packages that enables the use of LISC database.

[2]:
```
from lisc.objects.base import Base
from lisc.utils.db import SCDB, create_file_structure
from lisc import Counts
from lisc.utils.db import SCDB
from lisc.utils.io import save_object
from lisc.utils.io import load_object
from lisc.plts.counts import *
db = create_file_structure()
```

6.2 Data Wrangling

Here we will create terms that we will use to find articles with words that are most likely associated with them.

Here's an example of collecting terms and how LISC tells us which terms from the list is the most studied in their database

[3]:
```
terms = [['Hippocampus'], ['Cornu ammonis'], ['CA1'], ['CA3']]
counts = Counts()
counts.add_terms(terms)
counts.run_collection(verbose=True)
```

```
Running counts for:  Hippocampus
Running counts for:  Cornu ammonis
Running counts for:  CA1
Running counts for:  CA3
```

[4]:
```
print(counts.counts)
counts.check_top()
```

```
[[    0   790 16385  8126]
 [  790     0   529   238]
 [16385   529     0  6787]
 [ 8126   238  6787     0]]
The most studied term is  'Hippocampus'  with  107365  articles.
```

Now here's the actual terms that we will be using:

'terms_a' will be our primary terms list for lisc. We included CA1 and CA3 regions, as well as the hippocampus because we wanted to observe how unique the CA regions are.

‘terms_b’ will be our secondary terms list that will be compared to the primary terms list to find whether some of these terms are higher expressed for one region over the other based on the avalible liturature so far.

[5]:
```
terms_a = [['CA1'], ['CA3'], ['Hippocampus']]
terms_b = [['amygdala'], ['entorhinal cortex'], ['medial septum'],
           ['CA2'], ['DG'], ['mossy fibers'],['schaffer␣
 ↪collaterals'],['pyramidal cells'],
           ['LTP'], ['memory'], ['prefrontal cortex'],
           ['sensory'], ['spatial'], ['spikes']]
```

[6]:
```
counts.add_terms(terms_a, dim='A')
counts.add_terms(terms_b, dim='B')
```

```
Unloading previous terms words.
```

[7]:
```
counts.run_collection()
save_object(counts, 'tutorial_counts', directory=SCDB('lisc_db'))
counts = load_object('tutorial_counts', SCDB('lisc_db'))
```

[8]:
```
counts.check_data(data_type='counts', dim='A')
```

```
For  'CA1'          the highest association is  'memory'                with
5850
For  'CA3'          the highest association is  'memory'                with
2339
For  'Hippocampus'  the highest association is  'memory'                with
26753
```

[9]:
```
counts.check_data(data_type='counts', dim='B')
```

```
For  'amygdala'              the highest association is  'Hippocampus'  with
10580
For  'entorhinal cortex'     the highest association is  'Hippocampus'  with
3369
For  'medial septum'         the highest association is  'Hippocampus'  with
837
For  'CA2'                   the highest association is  'Hippocampus'  with
3433
For  'DG'                    the highest association is  'Hippocampus'  with
2538
For  'mossy fibers'          the highest association is  'Hippocampus'  with
650
For  'schaffer collaterals'  the highest association is  'CA1'          with
585
For  'pyramidal cells'       the highest association is  'Hippocampus'  with
3523
For  'LTP'                   the highest association is  'Hippocampus'  with
```

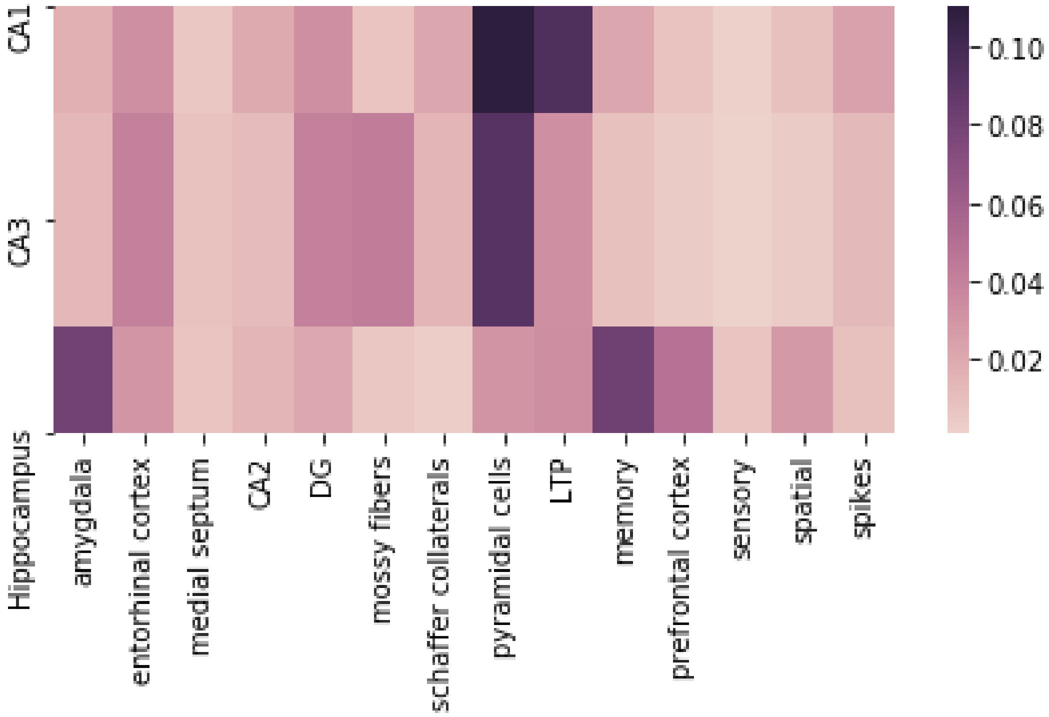
```
3927
For  'memory'                 the highest association is  'Hippocampus'  with
26753
For  'prefrontal cortex'      the highest association is  'Hippocampus'  with
6968
For  'sensory'                the highest association is  'Hippocampus'  with
2161
For  'spatial'                the highest association is  'Hippocampus'  with
11091
For  'spikes'                 the highest association is  'Hippocampus'  with
1218
```

[10]:
```
counts.compute_score('normalize', dim='A')
counts.compute_score('association')
```

As we can see below, the amount of literature collected from LISC display of specific terms pertaining to either CA1 and CA3 regions alone, sometimes both regions, and or are only mentioned when the Hippocamous is overall the core focus of the study. For example: * LTP was more commonly discussed in CA1 than CA3. * Pyramidal cells was more highly mentioned in CA1 over CA3 as well.

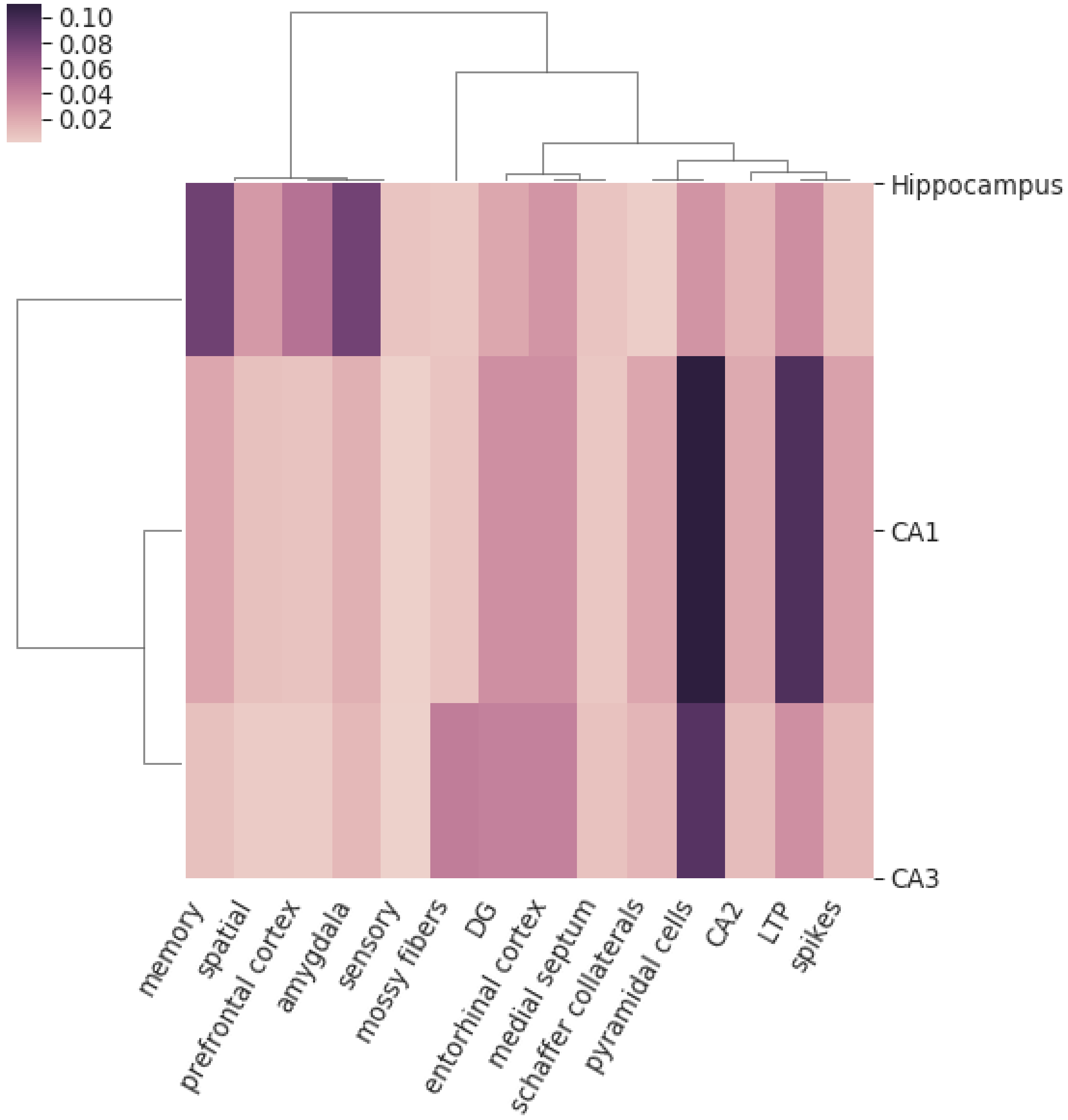
This suggests that there are indeed morphological and electrophysiological differences.

[11]:
```
plot_matrix(counts.score, counts.terms['B'].labels, counts.terms['A'].labels)
```



The clustermap gives a better visualization by using branches to show connections with secondary terms relating to each other that are ultimately found together under a primary term. For example: * LTP and spikes are highly mentioned under CA1, which then branches to the next connection to CA2. Because CA2 is next to CA1, it makes sense that this region would be mentioned along side these terms, as well as give us an idea that CA2 may have an important role with CA1 and the involvment of LTP and spikes. * The next branch connects to schaffer collaterals and pyramidal cells. This makes the first bullet point make more sense considering LTP is highly involved with the schaffer collaterals. Schaffer collaterals are found between CA1 and CA2, and in order to measure the activity of LTP and spikes between CA1 and CA2, then pyramidal cells will be the main cell to measure.

```
[12]: plot_clustermap(counts.score, counts.terms['B'].labels, counts.terms['A'].
      ↪labels)
```



7 Neuro Electro Dataset

To download data from NeuroElectro, we must first run this code below. This will download a zip file in your directory called "neuroelectro_data_dump.zip" containing the data needed to carry out the experiment.

[13]:
```
neuroelectro_dataset_url = "http://www.neuroelectro.org/static/src/
 ↪neuroelectro_data_download.zip"
handle = urlopen(neuroelectro_dataset_url)
data = handle.read()
fname = 'neuroelectro_data_dump.zip'
```

We tried to extract the 'neurophysiology_data.xlsx' programatically but grew too frustrated to do so, so we just did it manually through the computer's folder explorer.

[14]:
```
file_name = 'neurophysiology_data.xlsx'
df = read_excel(file_name)
```

We start by creating two dataframes: CA3_pyramidal_df and CA1_pyramidal_df. Here we can see all the variables in the dataframes that can be observed in Neuroelectro.

[15]:
```
df_data = df.set_index('NeuronType') #sets the index to the "NeuronType" column

df_data_filt = df_data[df_data.Species.isin(['Mice'])] #Filter's the dataframe␣
 ↪to only have Mice cell data

df_data_CA3 = df_data_filt.filter(regex = 'CA3', axis = 0) #Separates CA3 data␣
 ↪and creates a dataframe

df_data_CA1 = df_data_filt.filter(regex = 'CA1', axis = 0) #Separates CA1 data␣
 ↪and create a dataframe
```

[16]:
```
CA3_pyramidal_df = pd.DataFrame(df_data_CA3.loc['Hippocampus CA3 pyramidal␣
 ↪cell'])
CA1_pyramidal_df = pd.DataFrame(df_data_CA1.loc['Hippocampus CA1 pyramidal␣
 ↪cell'])
CA3_pyramidal_df.columns
```

[16]:
```
Index(['CellCapacitance', 'InputResistance', 'RestingMembranePotential',
       'MembraneTimeConstant', 'SpikeAmplitude', 'SpikeHalfWidth',
       'SpikeThreshold', 'Rheobase', 'FiringFrequency', 'AhpDuration',
       'CellDiameter', 'AccessResistance', 'SagRatio', 'SpikeOvershoot',
       'CellSurfaceArea', 'AhpAmplitude', 'FiSlope', 'SpontaneousFiringRate',
       'FastAhpAmplitude', 'FastAhpDuration', 'SlowAhpAmplitude',
       'SlowAhpDuration', 'SpikeWidth', 'AdpAmplitude', 'AdpDuration',
       'SpikePeak', 'AdaptationRatio', 'Species', 'Strain', 'ElectrodeType',
       'PrepType', 'JxnPotential', 'JxnOffset', 'Temp', 'Age', 'Weight',
       'Title', 'PubYear', 'PubmedLink', 'DataTableLinks', 'ArticleLink',
       'LastAuthor'],
      dtype='object')
```

[17]:

```
CA3_pyramidal_df.head()
```

[17]:

```
                                CellCapacitance  InputResistance  \
NeuronType
Hippocampus CA3 pyramidal cell            424.0            164.0
Hippocampus CA3 pyramidal cell              NaN            160.1

                                RestingMembranePotential  \
NeuronType
Hippocampus CA3 pyramidal cell                     -76.0
Hippocampus CA3 pyramidal cell                       NaN

                                MembraneTimeConstant  SpikeAmplitude  \
NeuronType
Hippocampus CA3 pyramidal cell                  61.0             NaN
Hippocampus CA3 pyramidal cell                  37.2             NaN

                                SpikeHalfWidth  SpikeThreshold  Rheobase  \
NeuronType
Hippocampus CA3 pyramidal cell            0.79           -58.0       NaN
Hippocampus CA3 pyramidal cell            1.18             NaN       NaN

                                FiringFrequency  AhpDuration  …  JxnOffset  \
NeuronType                                                    …
Hippocampus CA3 pyramidal cell              NaN          NaN  …        NaN
Hippocampus CA3 pyramidal cell              NaN         8.33  …        NaN

                                Temp   Age  Weight  \
NeuronType
Hippocampus CA3 pyramidal cell  31.0  42.0     NaN
Hippocampus CA3 pyramidal cell  29.0  21.5     NaN

Title  \
NeuronType
Hippocampus CA3 pyramidal cell  Stable mossy fiber long-term potentiation
requ…
Hippocampus CA3 pyramidal cell  Differential involvement of oriens/pyramidale
…

                                PubYear  \
NeuronType
```

```
Hippocampus CA3 pyramidal cell     2010
Hippocampus CA3 pyramidal cell     2005

                                                                  PubmedLink  \
NeuronType
Hippocampus CA3 pyramidal cell  http://www.ncbi.nlm.nih.gov/pubmed/20881117/
Hippocampus CA3 pyramidal cell  http://www.ncbi.nlm.nih.gov/pubmed/15486016/

                                                           DataTableLinks  \
NeuronType
Hippocampus CA3 pyramidal cell   http://neuroelectro.org/data_table/1721/
Hippocampus CA3 pyramidal cell  http://neuroelectro.org/data_table/10636/

                                                           ArticleLink  \
NeuronType
Hippocampus CA3 pyramidal cell  http://neuroelectro.org/article/27590/
Hippocampus CA3 pyramidal cell  http://neuroelectro.org/article/44738/

                                LastAuthor
NeuronType
Hippocampus CA3 pyramidal cell      Beck H
Hippocampus CA3 pyramidal cell     Buhl EH

[2 rows x 42 columns]
```

```
[18]: print("Hippocampus CA3 pyramidal cell (#, variables observed): ",df_data_CA3.
       ↪shape)
      print("Hippocampus CA1 pyramidal cell (#, variables observed): ",df_data_CA1.
       ↪shape)
```

```
Hippocampus CA3 pyramidal cell (#, variables observed):  (5, 42)
Hippocampus CA1 pyramidal cell (#, variables observed):  (13, 42)
```

Here we can see that there is only two Hippocampal CA3 pyramidal cells and eight Hippocampal CA1 pyramidal cells in Neuroelectro database. This is an issue so instead we decided to use all possible CA1 and CA3 observations to make comparisons.

```
[19]: print("All CA3 shape: ",df_data_CA3.shape)
      print("All CA1 shape: ",df_data_CA1.shape)
      df_data_CA1.head()
```

```
All CA3 shape:  (5, 42)
All CA1 shape:  (13, 42)
```

```
[19]:                                                     CellCapacitance  \
      NeuronType
      Hippocampus CA1 basket cell                                     NaN
      Hippocampus CA1 basket cell                                     NaN
      Hippocampus CA1 ivy neuron                                    107.0
      Hippocampus CA1 neurogliaform cell                             63.0
      Hippocampus CA1 oriens lacunosum moleculare neuron              NaN

                                                          InputResistance  \
      NeuronType
      Hippocampus CA1 basket cell                                   110.0
      Hippocampus CA1 basket cell                                   284.0
      Hippocampus CA1 ivy neuron                                    185.0
      Hippocampus CA1 neurogliaform cell                            239.5
      Hippocampus CA1 oriens lacunosum moleculare neuron            267.0

                                                          RestingMembranePotential  \
      NeuronType
      Hippocampus CA1 basket cell                                            -63.0
      Hippocampus CA1 basket cell                                            -59.8
      Hippocampus CA1 ivy neuron                                             -63.0
      Hippocampus CA1 neurogliaform cell                                     -63.5
      Hippocampus CA1 oriens lacunosum moleculare neuron                     -62.0

                                                          MembraneTimeConstant  \
      NeuronType
      Hippocampus CA1 basket cell                                         12.0
      Hippocampus CA1 basket cell                                         13.3
      Hippocampus CA1 ivy neuron                                          19.0
      Hippocampus CA1 neurogliaform cell                                  15.0
      Hippocampus CA1 oriens lacunosum moleculare neuron                  32.0

                                                          SpikeAmplitude  \
      NeuronType
      Hippocampus CA1 basket cell                                    NaN
      Hippocampus CA1 basket cell                                    NaN
      Hippocampus CA1 ivy neuron                                    56.0
      Hippocampus CA1 neurogliaform cell                            57.5
      Hippocampus CA1 oriens lacunosum moleculare neuron             NaN

                                                          SpikeHalfWidth  \
      NeuronType
      Hippocampus CA1 basket cell                                   0.40
      Hippocampus CA1 basket cell                                    NaN
      Hippocampus CA1 ivy neuron                                    0.80
      Hippocampus CA1 neurogliaform cell                            0.83
      Hippocampus CA1 oriens lacunosum moleculare neuron            0.80

                                                          SpikeThreshold  Rheobase  \
      NeuronType
      Hippocampus CA1 basket cell                                    NaN       NaN
      Hippocampus CA1 basket cell                                    NaN       NaN
      Hippocampus CA1 ivy neuron                                   -33.0       NaN
      Hippocampus CA1 neurogliaform cell                           -30.5       NaN
      Hippocampus CA1 oriens lacunosum moleculare neuron             NaN       NaN

                                                          FiringFrequency  \
      NeuronType
      Hippocampus CA1 basket cell                                     NaN
      Hippocampus CA1 basket cell                                   206.0
      Hippocampus CA1 ivy neuron                                      NaN
      Hippocampus CA1 neurogliaform cell                              NaN
      Hippocampus CA1 oriens lacunosum moleculare neuron              NaN

                                                          AhpDuration  …  \
      NeuronType                                                       …
      Hippocampus CA1 basket cell                                 NaN  …
      Hippocampus CA1 basket cell                                 NaN  …
      Hippocampus CA1 ivy neuron                                 73.0  …
      Hippocampus CA1 neurogliaform cell                         72.0  …
      Hippocampus CA1 oriens lacunosum moleculare neuron          NaN  …

                                                          JxnOffset  Temp   Age  \
      NeuronType
      Hippocampus CA1 basket cell                               NaN  32.0  12.0
      Hippocampus CA1 basket cell                               NaN  32.0  28.0
      Hippocampus CA1 ivy neuron                                NaN  33.0  17.5
      Hippocampus CA1 neurogliaform cell                        NaN  33.0  17.5
      Hippocampus CA1 oriens lacunosum moleculare neuron        NaN  32.0  12.0

                                                          Weight  \
      NeuronType
      Hippocampus CA1 basket cell                            NaN
      Hippocampus CA1 basket cell                            NaN
      Hippocampus CA1 ivy neuron                             NaN
      Hippocampus CA1 neurogliaform cell                     NaN
      Hippocampus CA1 oriens lacunosum moleculare neuron     NaN

                      Title  \
      NeuronType
      Hippocampus CA1 basket cell                         Transition to seizures in
      the isolated immatur…
      Hippocampus CA1 basket cell                         NMDA receptor-dependent
      long-term potentiation…
      Hippocampus CA1 ivy neuron                          Common origins of
      hippocampal Ivy and nitric o…
      Hippocampus CA1 neurogliaform cell                  Common origins of
      hippocampal Ivy and nitric o…
```

```
Hippocampus CA1 oriens lacunosum moleculare neuron  Transition to seizures in
the isolated immatur…

                                                    PubYear  \
NeuronType
Hippocampus CA1 basket cell                            2008
Hippocampus CA1 basket cell                            2007
Hippocampus CA1 ivy neuron                             2010
Hippocampus CA1 neurogliaform cell                     2010
Hippocampus CA1 oriens lacunosum moleculare neuron     2008

      PubmedLink  \
NeuronType
Hippocampus CA1 basket cell
http://www.ncbi.nlm.nih.gov/pubmed/17991696/
Hippocampus CA1 basket cell
http://www.ncbi.nlm.nih.gov/pubmed/17884930/
Hippocampus CA1 ivy neuron
http://www.ncbi.nlm.nih.gov/pubmed/20147544/
Hippocampus CA1 neurogliaform cell
http://www.ncbi.nlm.nih.gov/pubmed/20147544/
Hippocampus CA1 oriens lacunosum moleculare neuron
http://www.ncbi.nlm.nih.gov/pubmed/17991696/

DataTableLinks  \
NeuronType
Hippocampus CA1 basket cell
http://neuroelectro.org/data_table/8047/
Hippocampus CA1 basket cell
http://neuroelectro.org/data_table/11371/
Hippocampus CA1 ivy neuron
http://neuroelectro.org/data_table/1838/
Hippocampus CA1 neurogliaform cell
http://neuroelectro.org/data_table/1838/
Hippocampus CA1 oriens lacunosum moleculare neuron
http://neuroelectro.org/data_table/8047/

ArticleLink  \
NeuronType
Hippocampus CA1 basket cell
http://neuroelectro.org/article/11195/
Hippocampus CA1 basket cell
http://neuroelectro.org/article/46287/
Hippocampus CA1 ivy neuron
http://neuroelectro.org/article/35516/
Hippocampus CA1 neurogliaform cell
http://neuroelectro.org/article/35516/
```

```
Hippocampus CA1 oriens lacunosum moleculare neuron
http://neuroelectro.org/article/11195/

                                                      LastAuthor
NeuronType
Hippocampus CA1 basket cell                            Carlen PL
Hippocampus CA1 basket cell                          Kullmann DM
Hippocampus CA1 ivy neuron                             McBain CJ
Hippocampus CA1 neurogliaform cell                     McBain CJ
Hippocampus CA1 oriens lacunosum moleculare neuron     Carlen PL

[5 rows x 42 columns]
```

It looks like we have gathered more observations for CA3 and CA1, but not enough. Unfortunately we have too many NaN values. Using df.dropna() method, will delete everything, so we will now use 3 variables that have the most data. This will produce a cleaner version of CA1 and CA3 dataframes from Neuroelectro.

[20]:

```
Age_list1 = []
IR_list1 = []
NeuronType_list1 = []
Resting_Membrane_PotentialList = []

for i in range(0, len(df_data_CA1['Age'])):
    Age_list1.append(df_data_CA1['Age'].iloc[i])
for i in range(0, len(df_data_CA1['InputResistance'])):
    IR_list1.append(df_data_CA1['InputResistance'].iloc[i])
for i in range(0, len(df_data_CA1['RestingMembranePotential'])):
    Resting_Membrane_PotentialList.
 ↪append(df_data_CA1['RestingMembranePotential'].iloc[i])
for item in df_data_CA1.index:
    NeuronType_list1.append(item)

dict1 = {
    'Age' : Age_list1,
    'Input Res' : IR_list1,
    'Resting Membrane Potential' : Resting_Membrane_PotentialList
}

CA1_neuroelectro = pd.DataFrame(dict1)

CA1_neuroelectro = CA1_neuroelectro.dropna()

CA1_neuroelectro['Type'] = 'CA1'

CA1_neuroelectro.head()
```

```
Age_list1 = []
IR_list1 = []
NeuronType_list1 = []
Resting_Membrane_PotentialList = []

for i in range(0, len(df_data_CA1['Age'])):
    Age_list1.append(df_data_CA1['Age'].iloc[i])
for i in range(0, len(df_data_CA1['InputResistance'])):
    IR_list1.append(df_data_CA1['InputResistance'].iloc[i])
for i in range(0, len(df_data_CA1['RestingMembranePotential'])):
    Resting_Membrane_PotentialList.
 ↪append(df_data_CA1['RestingMembranePotential'].iloc[i])
for item in df_data_CA1.index:
    NeuronType_list1.append(item)

dict1 = {
    'Age' : Age_list1,
    'Input Res' : IR_list1,
    'Resting Membrane Potential' : Resting_Membrane_PotentialList
}

CA3_neuroelectro = pd.DataFrame(dict1)

CA3_neuroelectro = CA1_neuroelectro.dropna()

CA3_neuroelectro['Type'] = 'CA3'
```

After organizing our data, we need all the dataframes in one collection so we concatinated them together in order to run a covariance matrix

[21]:
```
CA = pd.concat([CA1_neuroelectro,CA3_neuroelectro])
CA = CA.reset_index(drop=True)
CA.head()
```

[21]:
```
    Age  Input Res  Resting Membrane Potential Type
0  12.0      110.0                       -63.0  CA1
1  28.0      284.0                       -59.8  CA1
2  17.5      185.0                       -63.0  CA1
3  17.5      239.5                       -63.5  CA1
4  12.0      267.0                       -62.0  CA1
```

….quick cleaning and normalization….

[22]:
```
xCA = CA.drop('Type',axis=1)
xCA = (xCA - xCA.mean())/xCA.std()
```

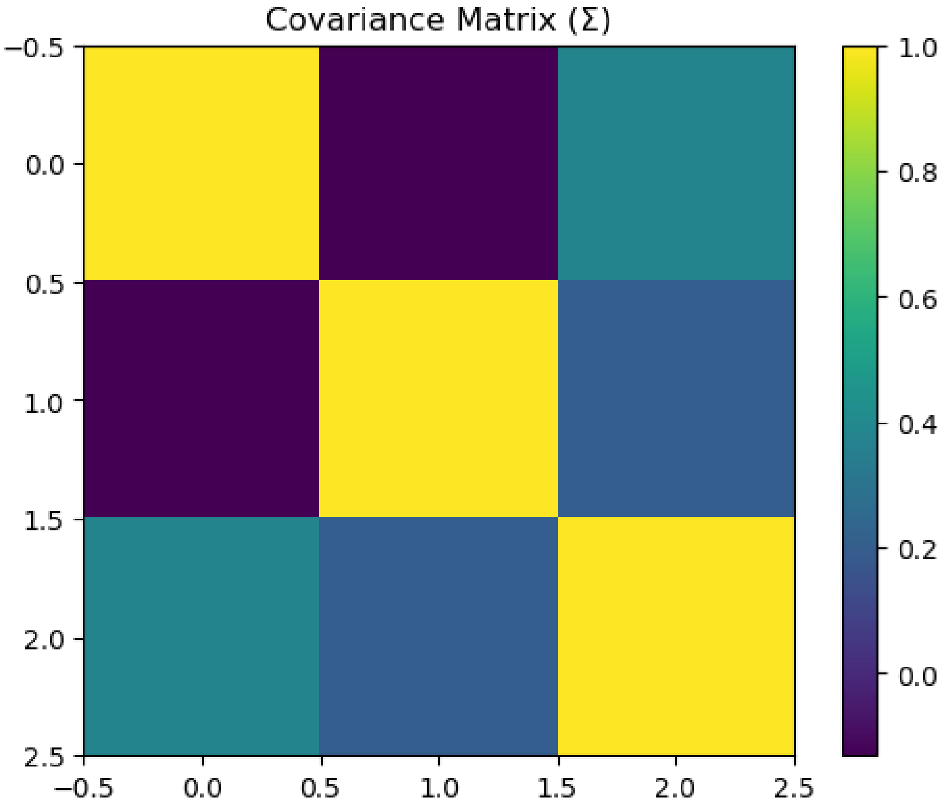
[23]:

```
#Restores our plot to its default setting (from using LISC)
mpl.rcParams.update(mpl.rcParamsDefault)
```

Once we had everything set, we ran our data through a covariance method and plotted it in a matrix as well as a the calculated eiganvalues.

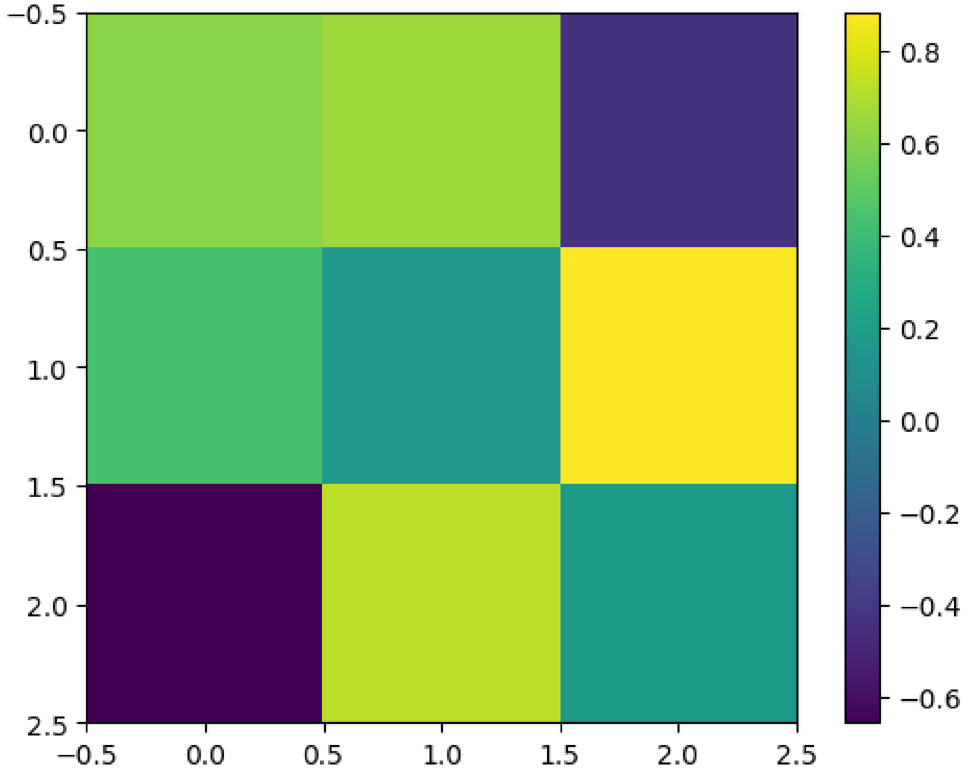
[24]:

```
xCAcov = xCA.cov()
plt.imshow(xCAcov)
plt.colorbar()
plt.title('Covariance Matrix (\u03A3)')
plt.show()
```



[25]:

```
eigenvectors, eigenvalues = np.linalg.eig(xCAcov)

plt.imshow(eigenvalues)
plt.colorbar()
plt.show()
```



Seeing how this dataset is not going to be the most useful for our exploratory analysis, we looked past NeuroElectro and analyzed Neuropixels data instead. Let's see what we can unpack...

[26]:
```
manifest_path = '/datasets/allen-brain-observatory/visual-coding-neuropixels/
 ↪ecephys-cache/manifest.json'
cache = EcephysProjectCache.fixed(manifest=manifest_path)
```

Using the same methods and techniques as NeuroElectro, we ran the same code except on this new dataset

[27]:
```
units = cache.get_units()
CA1_units = units[units['ecephys_structure_acronym']=='CA1']
CA3_units = units[units['ecephys_structure_acronym']=='CA3']
CA1_wt= CA1_units[CA1_units['genotype']=='wt/wt']
CA3_wt= CA3_units[CA3_units['genotype']=='wt/wt']
```

Again we removed any non-integer values in our dataframes and concatinated them together. We also filtered out Wild type (wt/wt) in order to reduce genotype as a factor.

[28]:
```
CA1=CA1_wt.drop(columns=['name',␣
 ↪'has_lfp_data','date_of_acquisition','published_at','ecephys_session_id',
                         ␣
 ↪'ecephys_structure_id','specimen_id','phase','session_type','sex','genotype'])
```

```
CA3=CA3_wt.drop(columns=['name',␣
↪'has_lfp_data','date_of_acquisition','published_at','ecephys_session_id',
                         ␣
↪'ecephys_structure_id','specimen_id','phase','session_type','sex','genotype'])
CA1.dropna()
CA3.dropna()
```

[28]:

```
           waveform_PT_ratio  waveform_amplitude  amplitude_cutoff  \
id
950926721           0.370461           62.947755          0.015842
950926765           0.247989           75.366135          0.067578
950926788           0.366901           68.652480          0.074152
950926843           0.423820           92.673165          0.000572
950926862           0.629321          175.262685          0.000025
…                      …                 …               …
951892251           0.337831          108.793815          0.025387
951892264           0.405969          178.746555          0.000459
951894251           0.408772          211.770195          0.003394
951892278           0.303730          100.360650          0.002202
951892285           0.655855          149.559735          0.003746

           cumulative_drift    d_prime  waveform_duration  ecephys_channel_id  \
id
950926721            186.54   3.461425           0.439531           849994620
950926765            169.86   2.478424           0.288442           849994620
950926788            158.79   3.052954           0.274707           849994620
950926843            132.49   5.621975           0.247236           849994622
950926862            111.27  10.058965           0.315913           849994632
…                     …        …                 …                 …
951892251            240.30   3.669196           0.741709           850083074
951892264            163.81   6.300854           0.576884           850083074
951894251            152.50   6.227688           0.618090           850083074
951892278            122.96   4.554299           0.206030           850083078
951892285            228.14   3.327682           0.219765           850083080

           firing_rate  waveform_halfwidth  isi_violations  …  local_index  \
id                                                          …
950926721     8.941300            0.192295        0.073100  …           10
950926765     8.581566            0.164824        0.098939  …           10
950926788    17.913986            0.164824        0.011234  …           10
950926843    29.295764            0.123618        0.006412  …           11
950926862    12.154299            0.151089        0.000000  …           16
…                 …                 …             …  …          …
951892251    11.361517            0.233501        0.449293  …          126
951892264     2.578912            0.206030        0.065813  …          126
951894251     2.392841            0.178559        0.019112  …          126
951892278    33.948794            0.082412        0.029999  …          128
```

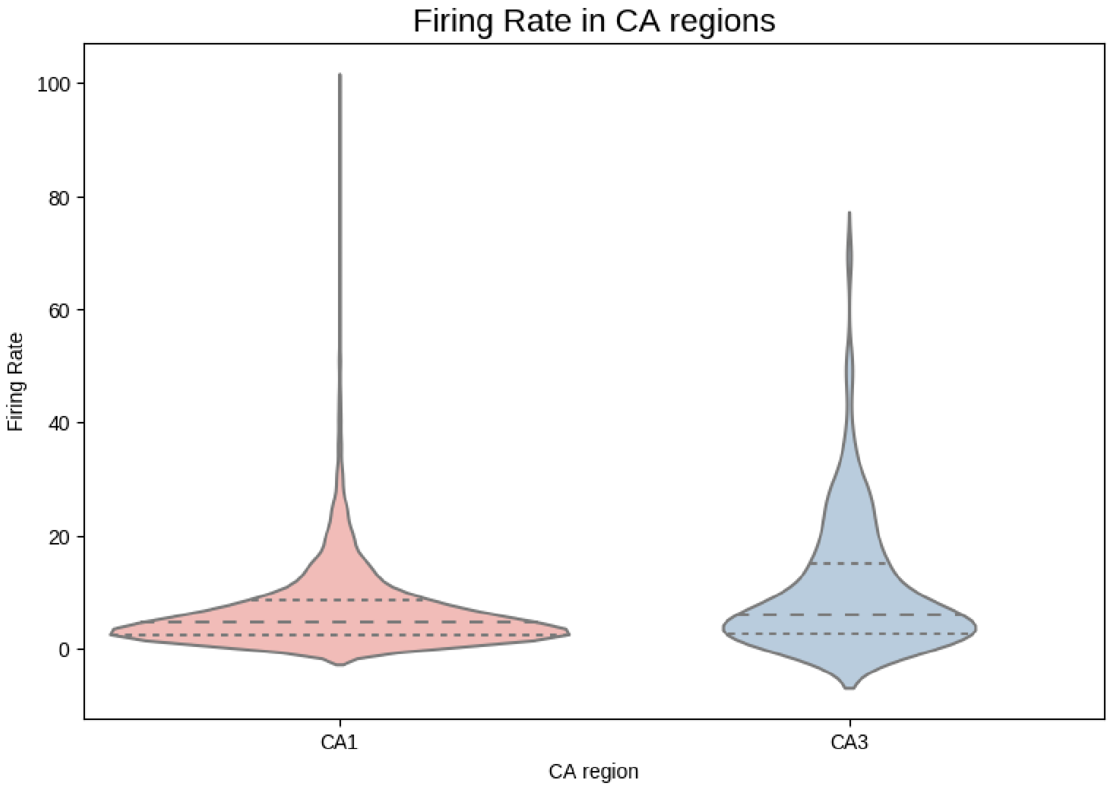
```
951892285    29.921799            0.137353        0.029171  …          129

           probe_horizontal_position  probe_vertical_position  \
id
950926721                         59                      120
950926765                         59                      120
950926788                         59                      120
950926843                         27                      120
950926862                         43                      180
…                              …                      …
951892251                         59                     1280
951892264                         59                     1280
951894251                         59                     1280
951892278                         43                     1300
951892285                         11                     1300

           anterior_posterior_ccf_coordinate  dorsal_ventral_ccf_coordinate  \
id
950926721                               7731                           3313
950926765                               7731                           3313
950926788                               7731                           3313
950926843                               7734                           3308
950926862                               7753                           3274
…                                      …                            …
951892251                               8124                           3240
951892264                               8124                           3240
951894251                               8124                           3240
951892278                               8126                           3222
951892285                               8127                           3204

           left_right_ccf_coordinate  ecephys_structure_acronym  \
id
950926721                       9008                        CA3
950926765                       9008                        CA3
950926788                       9008                        CA3
950926843                       9011                        CA3
950926862                       9025                        CA3
…                              …                        …
951892251                       8675                        CA3
951892264                       8675                        CA3
951894251                       8675                        CA3
951892278                       8685                        CA3
951892285                       8695                        CA3

           lfp_sampling_rate  sampling_rate  age_in_days
id
950926721        1249.996436   29999.914455        142.0
```

```
950926765         1249.996436   29999.914455         142.0
950926788         1249.996436   29999.914455         142.0
950926843         1249.996436   29999.914455         142.0
950926862         1249.996436   29999.914455         142.0
…                       …             …          …
951892251         1249.999868   29999.996830         119.0
951892264         1249.999868   29999.996830         119.0
951894251         1249.999868   29999.996830         119.0
951892278         1249.999868   29999.996830         119.0
951892285         1249.999868   29999.996830         119.0

[376 rows x 34 columns]
```

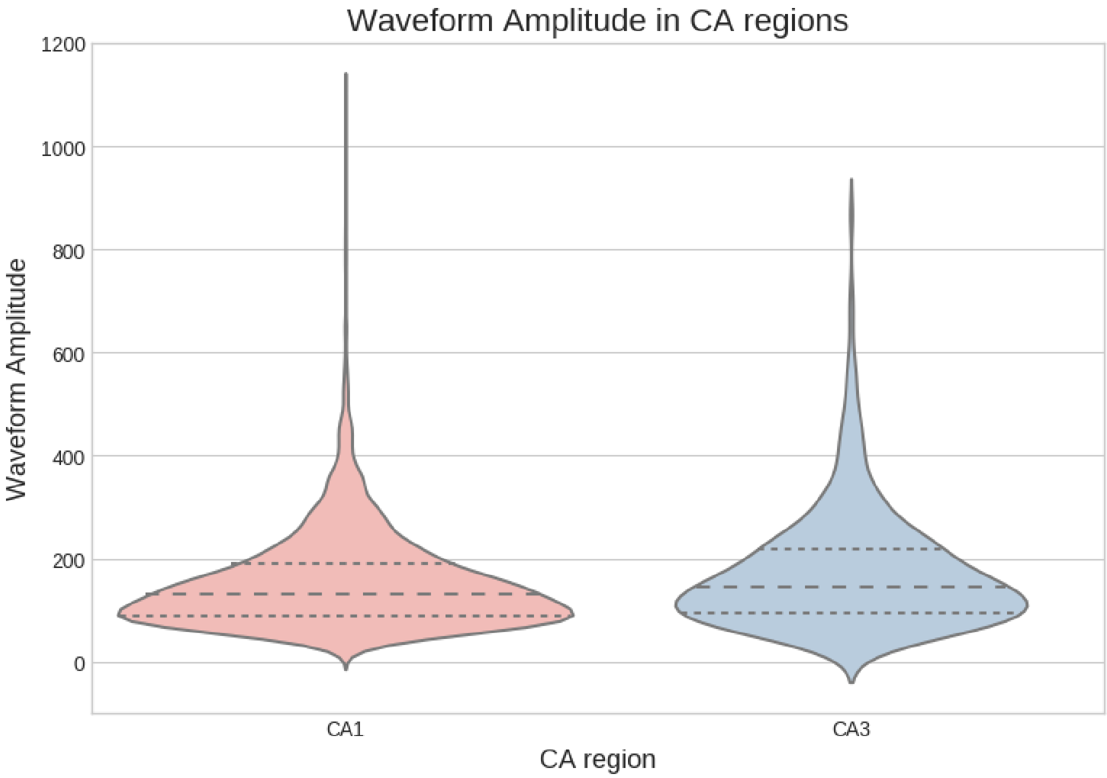
```
[29]: CA= pd.concat([CA1,CA3])
      CA= CA.reset_index(drop=True)
      CA = CA.dropna() #will remove any columns or rows with NaN values
```

This will create violin plots which are very similar to box plots. They will compare firing rate, waveform amplitude, and waveform repolarization between CA1 and CA3 regions.

```
[30]: fig, ax= plt.subplots(figsize=(9,6))
      ax = sns.violinplot(x="ecephys_structure_acronym", y="firing_rate",␣
       ↪inner='quartile', data=CA,palette='Pastel1',
                          width=0.9)
      ax.set_title('Firing Rate in CA regions', fontsize=16);
      ax.set_ylabel('Firing Rate')
      ax.set_xlabel('CA region')
      plt.rc('axes', labelsize=13)
      plt.style.use('seaborn-whitegrid')
      plt.show()
```

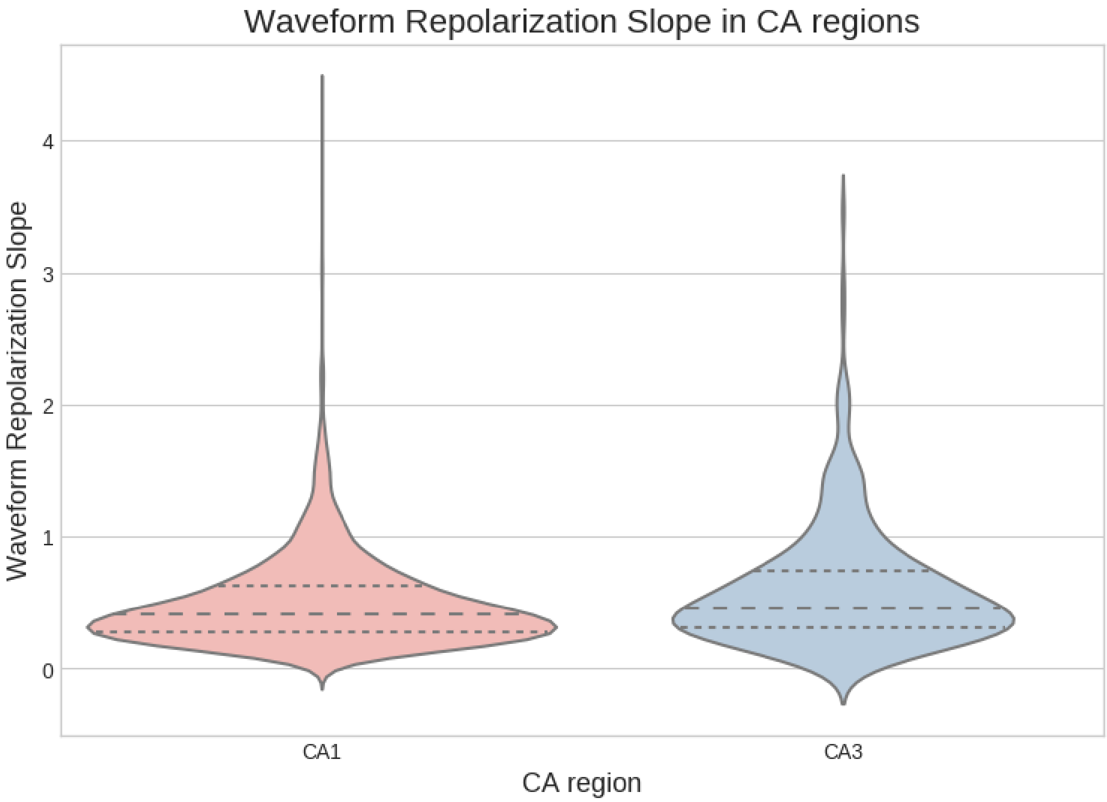


```
[31]: fig, ax= plt.subplots(figsize=(9,6))
      ax = sns.violinplot(x="ecephys_structure_acronym", y="waveform_amplitude",␣
       ↪inner='quartile', data=CA,palette='Pastel1',
                          width=0.9)
      ax.set_title('Waveform Amplitude in CA regions', fontsize=16);
      ax.set_ylabel('Waveform Amplitude')
      ax.set_xlabel('CA region')
      plt.rc('axes', labelsize=13)
      plt.style.use('seaborn-whitegrid')
      plt.show()
```



[32]:

```
fig, ax= plt.subplots(figsize=(9,6))
ax = sns.violinplot(x="ecephys_structure_acronym",␣
 ↪y="waveform_repolarization_slope", inner='quartile',␣
 ↪data=CA,palette='Pastel1',
                    width=0.9)
ax.set_title('Waveform Repolarization Slope in CA regions', fontsize=16);
ax.set_ylabel('Waveform Repolarization Slope')
ax.set_xlabel('CA region')
plt.rc('axes', labelsize=13)
plt.style.use('seaborn-whitegrid')
plt.show()
```



8 Covariance Matrix

By creating a covariance matrix, we seek to observe possible variables (columns) that may have an influence on other variables. Now to begin our covariance analysis, we must first normalize our data.

```
[33]: xCA = CA.drop('ecephys_structure_acronym',axis=1)
      xCA = (xCA - xCA.mean())/xCA.std()
      xCA.head()
```

```
[33]:    waveform_PT_ratio  waveform_amplitude  amplitude_cutoff  cumulative_drift  \
      0           0.414752            0.529701          1.285532         -0.832534
      1           0.015588            0.573607         -0.637070         -1.000531
      2           0.326178            1.065052          1.296322         -0.723090
      3          -0.058233           -0.240202          1.264721         -0.510117
      4          -0.192598           -0.031705          0.505544         -0.330376

          d_prime  waveform_duration  ecephys_channel_id  firing_rate  \
      0 -0.235038          -0.327165           -0.232658     0.291787
      1  0.087323           0.685172           -0.232658     1.059359
```

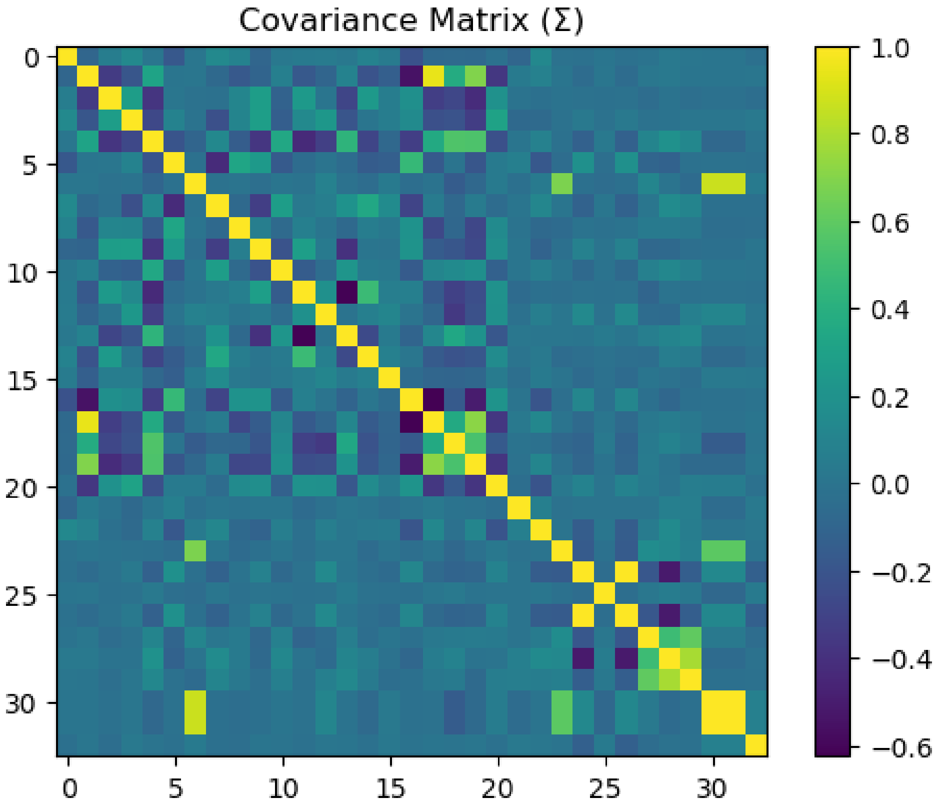
```
2 -0.515366           0.685172          -0.232658    -0.580517
3 -0.347554           1.978713          -0.232658     1.610371
4 -0.569270          -0.045961          -0.232658    -0.242299

   waveform_halfwidth  isi_violations  …  ecephys_probe_id  local_index  \
0            0.513330       -0.432318  …          -0.50672    -0.930037
1            0.896424       -0.847732  …          -0.50672    -0.930037
2            1.534913       -0.432678  …          -0.50672    -0.930037
3            1.407215       -0.672942  …          -0.50672    -0.930037
4            0.896424        0.186171  …          -0.50672    -0.930037

   probe_horizontal_position  probe_vertical_position  \
0                   -1.19343                 -0.93969
1                   -1.19343                 -0.93969
2                   -1.19343                 -0.93969
3                   -1.19343                 -0.93969
4                   -1.19343                 -0.93969

   anterior_posterior_ccf_coordinate  dorsal_ventral_ccf_coordinate  \
0                          -0.245882                      -0.430815
1                          -0.245882                      -0.430815
2                          -0.245882                      -0.430815
3                          -0.245882                      -0.430815
4                          -0.245882                      -0.430815

   left_right_ccf_coordinate  lfp_sampling_rate  sampling_rate  age_in_days
0                   -1.47016          -0.254096      -0.254096     1.473235
1                   -1.47016          -0.254096      -0.254096     1.473235
2                   -1.47016          -0.254096      -0.254096     1.473235
3                   -1.47016          -0.254096      -0.254096     1.473235
4                   -1.47016          -0.254096      -0.254096     1.473235

[5 rows x 33 columns]
```

[34]:

```python
#Restores our plot to its default setting (from using LISC)
mpl.rcParams.update(mpl.rcParamsDefault)
```

[35]:

```python
xCAcov = xCA.cov()
plt.imshow(xCAcov)
plt.colorbar()
plt.title('Covariance Matrix (\u03A3)')
plt.show()
```



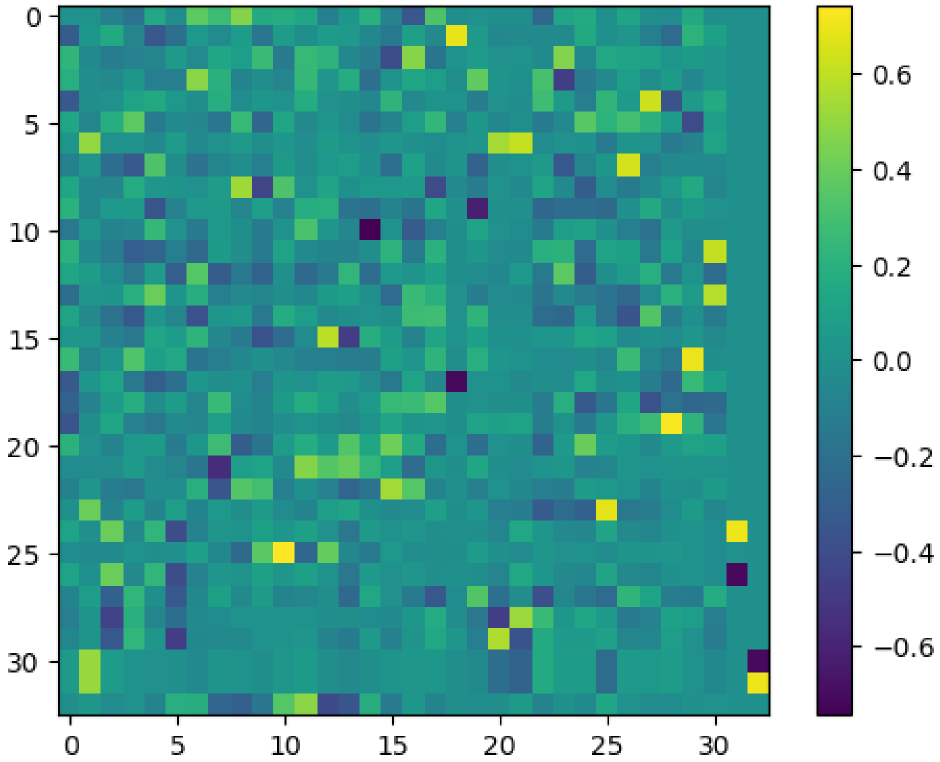
From this we can see from the yellow squares, which indicates a possible relationship. From this we can see that amplitude cutoff vs repolarization and recovery slope have a relationship.

9 Eigenvalues, eiganvectors, and eigansum

As a quick reminder, an eigenvalue is a number, telling you how much variance there is in the data in its corresponding eigenvector, and how spread out the data is on the line. The eigenvector with the highest eigenvalue is therefore the principal component. the amount of eigenvectors/values that exist equals the number of dimensions the data set has. These directions are where there is most variation, and that is where there is more information (think about this the reverse way round. If there was no variation in the data there would be no information).

We continued this analysis by plotting the eiganvalues calculated from the dataframe

[36]:
```
eigenvectors, eigenvalues = np.linalg.eig(xCAcov)

plt.imshow(eigenvalues)
plt.colorbar()
plt.show()
```



[37]:

```
eigenSum = np.sum(eigenvectors)
```

To analyze our data in a more effecient manner, we need to reduce the dimensions from 33 to something easier to visualize. We used PCA to reduce the dimensions from 33 to 2 and hopefully exactract any features of interest.

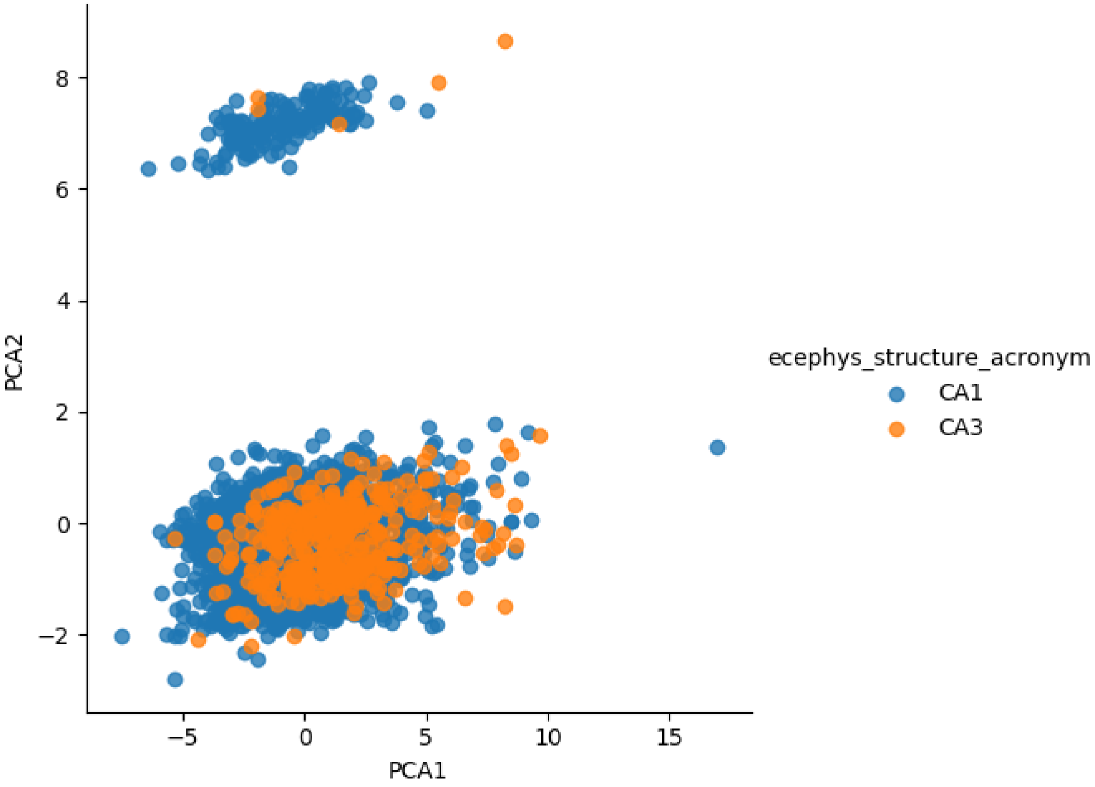
Here we can see that the PCA has separated the cells into two clusters, not necessarily by cell type as indicated with two different colors. To get a greater insight on what these clusters represent, we need to take a Kmeans clustering on the data. We will attampt to fit our 2 dimensional data into a kmeans model with 2 clusters and see what our data looks like.

We start to take a deeper look into reducing the dimentions of the data through PCA and transform it to two dimensions

[38]:

```
pca = PCA(n_components = 2)
X_2D = pca.fit_transform(xCA)

CA['PCA1'] = X_2D[:, 0]
CA['PCA2'] = X_2D[:, 1]

a = CA['PCA1']
b = CA['PCA2']
```

```
sns.lmplot('PCA1', 'PCA2', hue='ecephys_structure_acronym', data=CA,␣
 ↪fit_reg=False)
plt.show()
```

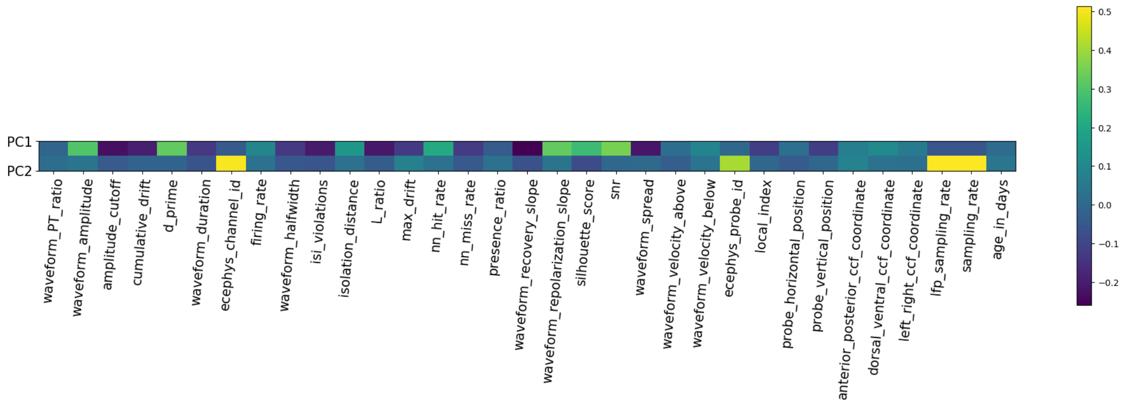


[39]:
```
print("original shape:   ", xCA.shape)
print("transformed shape:", X_2D.shape)
```

```
original shape:    (3370, 33)
transformed shape: (3370, 2)
```

After plotting our PCA values, we can see there is a cluster of data separated by PCA2. To further investigate, we must take a look at the PCA components for futher insight.

[40]:
```
plt.figure(figsize=(20,5))
plt.imshow(pca.components_,cmap='viridis',)
plt.yticks([0,1],['PC1','PC2'],fontsize=15)
plt.colorbar(orientation='vertical')
plt.tight_layout()
plt.xticks(range(len(xCA.columns)),xCA.columns,rotation=85, fontsize = 15)

plt.show()
```



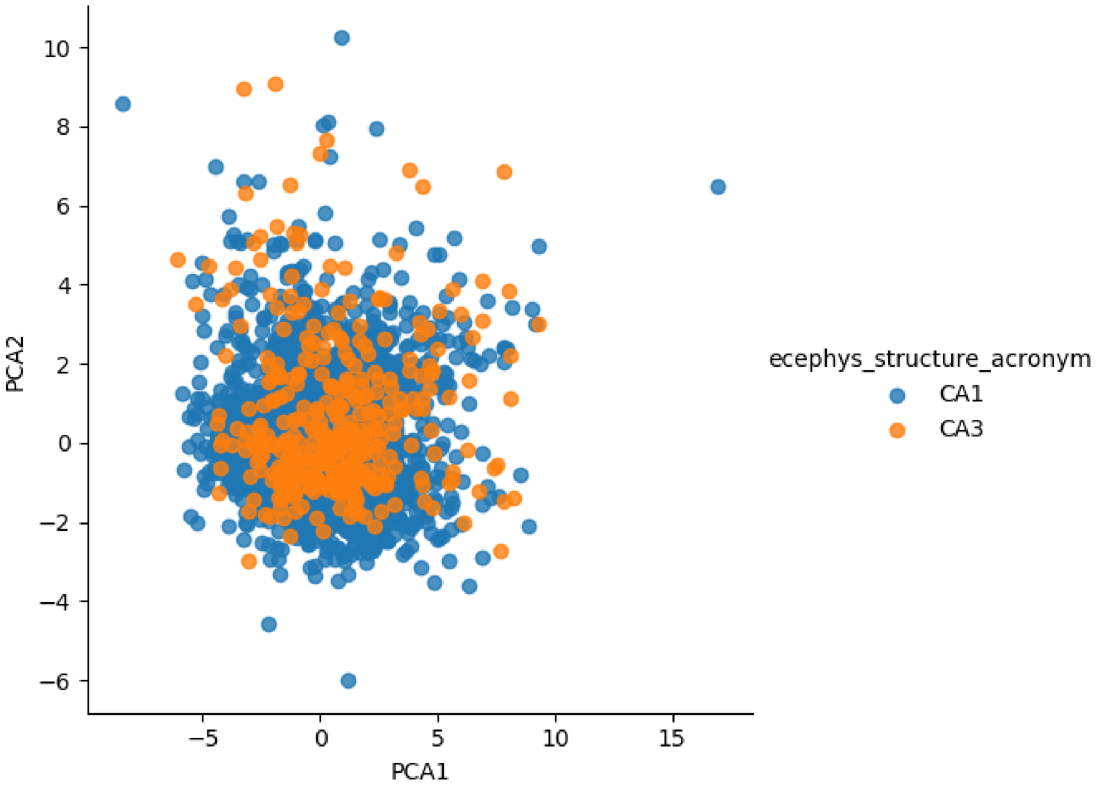
So although we dropped our non integer values from our data, it turns out there are some integer values that are not necessarily electrophysiological measurements of the cells. We gotta get rid of them and run it again!

```
[41]: xCA = xCA.drop('ecephys_channel_id',axis=1)
      xCA = xCA.drop('lfp_sampling_rate',axis=1)
      xCA = xCA.drop('sampling_rate',axis=1)
      xCA = xCA.drop('ecephys_probe_id',axis=1)
      xCA = xCA.drop('local_index',axis=1)
      xCA = xCA.drop('probe_horizontal_position',axis=1)
      xCA = xCA.drop('probe_vertical_position',axis=1)
      xCA = xCA.drop('anterior_posterior_ccf_coordinate',axis=1)
      xCA = xCA.drop('dorsal_ventral_ccf_coordinate',axis=1)
      xCA = xCA.drop('left_right_ccf_coordinate',axis=1)



      xCA.head()
```

```
[41]:    waveform_PT_ratio  waveform_amplitude  amplitude_cutoff  cumulative_drift  \
      0           0.414752            0.529701          1.285532         -0.832534
      1           0.015588            0.573607         -0.637070         -1.000531
      2           0.326178            1.065052          1.296322         -0.723090
      3          -0.058233           -0.240202          1.264721         -0.510117
      4          -0.192598           -0.031705          0.505544         -0.330376

          d_prime  waveform_duration  firing_rate  waveform_halfwidth  \
      0 -0.235038          -0.327165     0.291787            0.513330
      1  0.087323           0.685172     1.059359            0.896424
      2 -0.515366           0.685172    -0.580517            1.534913
      3 -0.347554           1.978713     1.610371            1.407215
      4 -0.569270          -0.045961    -0.242299            0.896424

         isi_violations  isolation_distance  …  nn_miss_rate  presence_ratio  \
      0       -0.432318           -0.244700  …      0.355459        0.278524
      1       -0.847732           -0.205016  …      0.609522        0.278524
      2       -0.432678           -0.612858  …     -0.331092        0.278524
      3       -0.672942           -0.138802  …      0.827865        0.278524
      4        0.186171           -0.444595  …      0.071025        0.278524

         waveform_recovery_slope  waveform_repolarization_slope  silhouette_score  \
      0                -0.846380                       0.158400          0.517975
      1                -0.174460                      -0.377538          1.203822
      2                -0.281136                      -0.324268          0.002485
      3                 0.410489                      -0.788778         -0.258724
      4                 0.702341                      -0.227823         -1.328843

              snr  waveform_spread  waveform_velocity_above  \
      0  0.304971        -0.919645                 1.709568
      1  0.647027        -1.309138                 2.725270
      2  0.609579        -0.919645                 2.386703
      3 -0.613775        -1.309138                 4.756676
      4 -0.446970        -0.919645                 1.371000

         waveform_velocity_below  age_in_days
      0                -0.012490     1.473235
      1                -2.225728     1.473235
      2                -0.908324     1.473235
      3                -4.597054     1.473235
      4                -0.012490     1.473235

      [5 rows x 23 columns]
```

[42]:

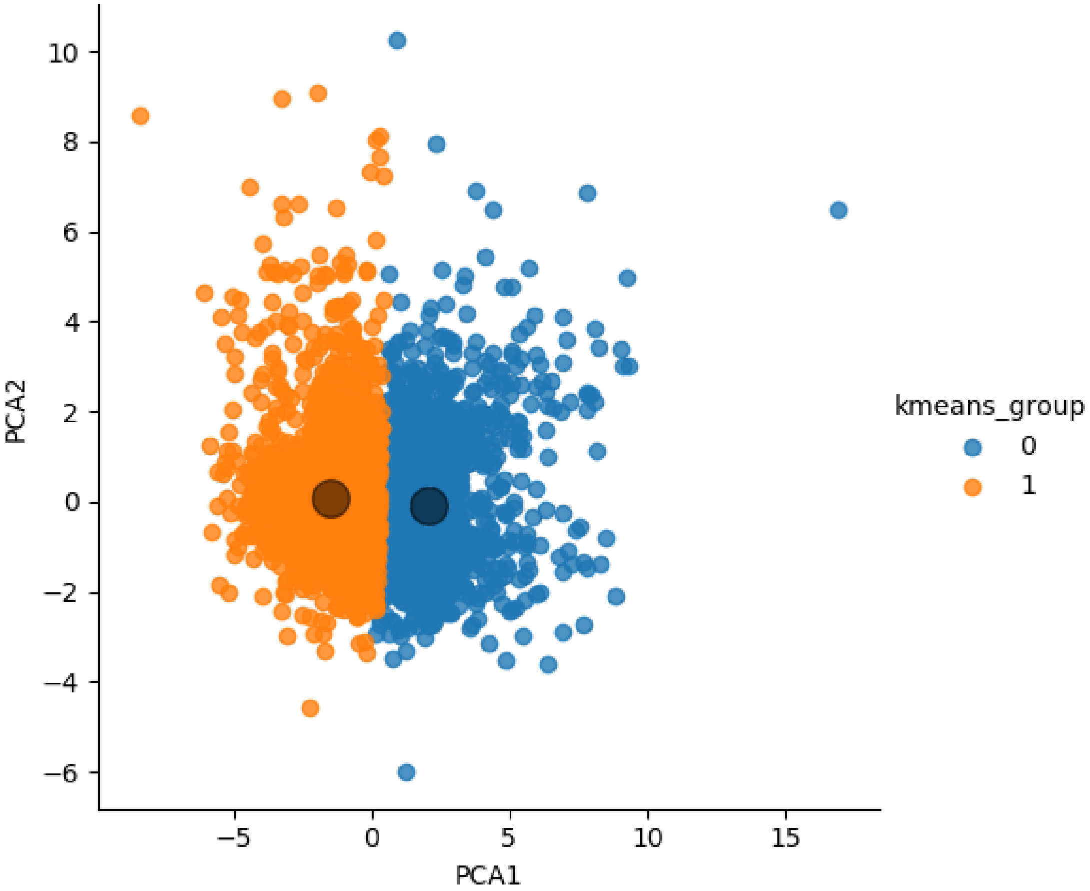
```
pca = PCA(n_components = 2)
X_2D = pca.fit_transform(xCA)

CA['PCA1'] = X_2D[:, 0]
CA['PCA2'] = X_2D[:, 1]



a = CA['PCA1']
b = CA['PCA2']

sns.lmplot('PCA1', 'PCA2', hue='ecephys_structure_acronym', data=CA,␣
 ↪fit_reg=False)
plt.show()
```



Unfortunately creating PCA's did not reveal any significant features in our data.

[44]:
```
kmeans2 = KMeans(n_clusters=2)
kmeans2.fit(X_2D)
y_kmeans2 = kmeans2.predict(X_2D)
```

[45]:
```
CA['kmeans_group'] = y_kmeans2

sns.lmplot("PCA1", "PCA2", hue='kmeans_group', data=CA, fit_reg=False)
centers2 = kmeans2.cluster_centers_
plt.scatter(centers2[:, 0], centers2[:, 1], c='black', s=200, alpha=0.5)
plt.show()
```



The appearance of this k means clustering is a bit weird as it is splitting our data right in half based on some kind of cellular type or characteristic. We notice how the colors change depending on what side of the PCA1 the cell is on. If it has a positive PCA1 value, the cell is orange. If the cell has a negative PCA1 value, the cell is labeled as blue.

9.0.1 The following displays exploratory analyses where we created plots and calculated statistics to determine whether or not there was a relationship between amplitude, waveform recovery slope, and amplitude cutoff.

[46]:
```
CA['ecephys_structure_acronym'].unique()
```

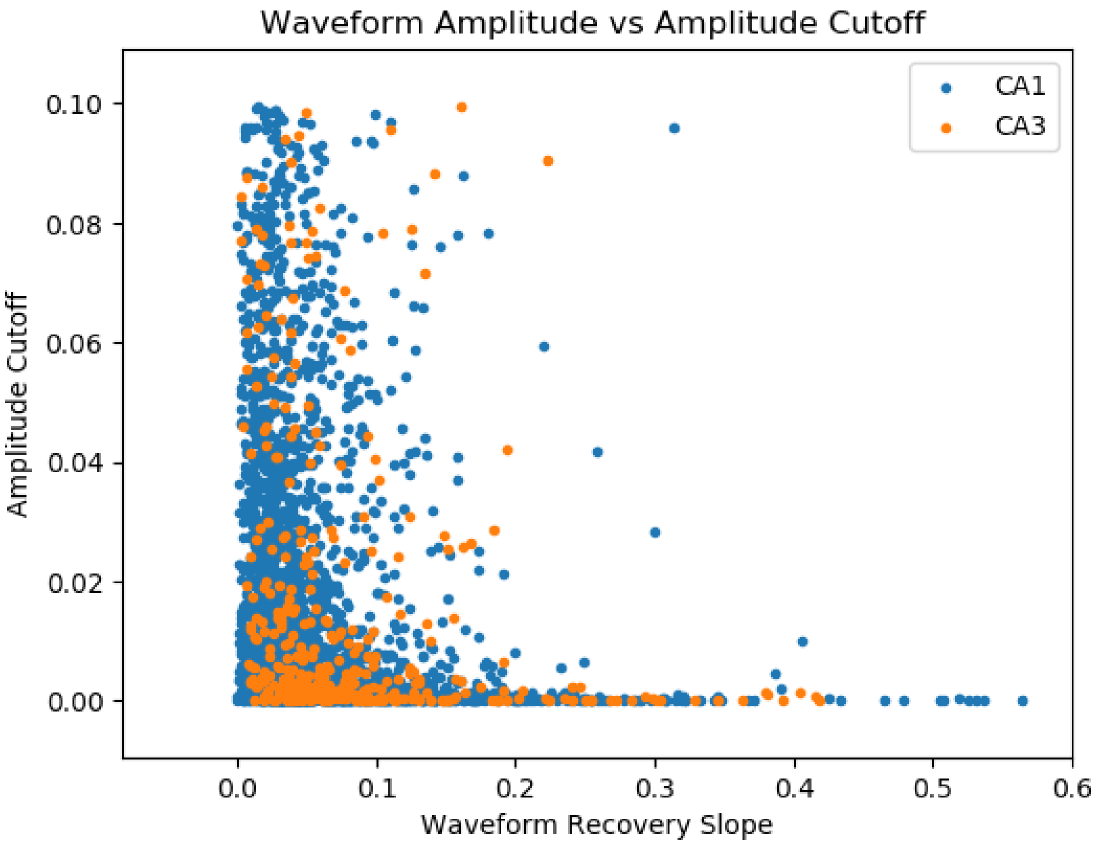
[46]:
```
array(['CA1', 'CA3'], dtype=object)
```

[47]:
```
#Restores our plot to its default setting (from using LISC)
mpl.rcParams.update(mpl.rcParamsDefault)
```

[133]:

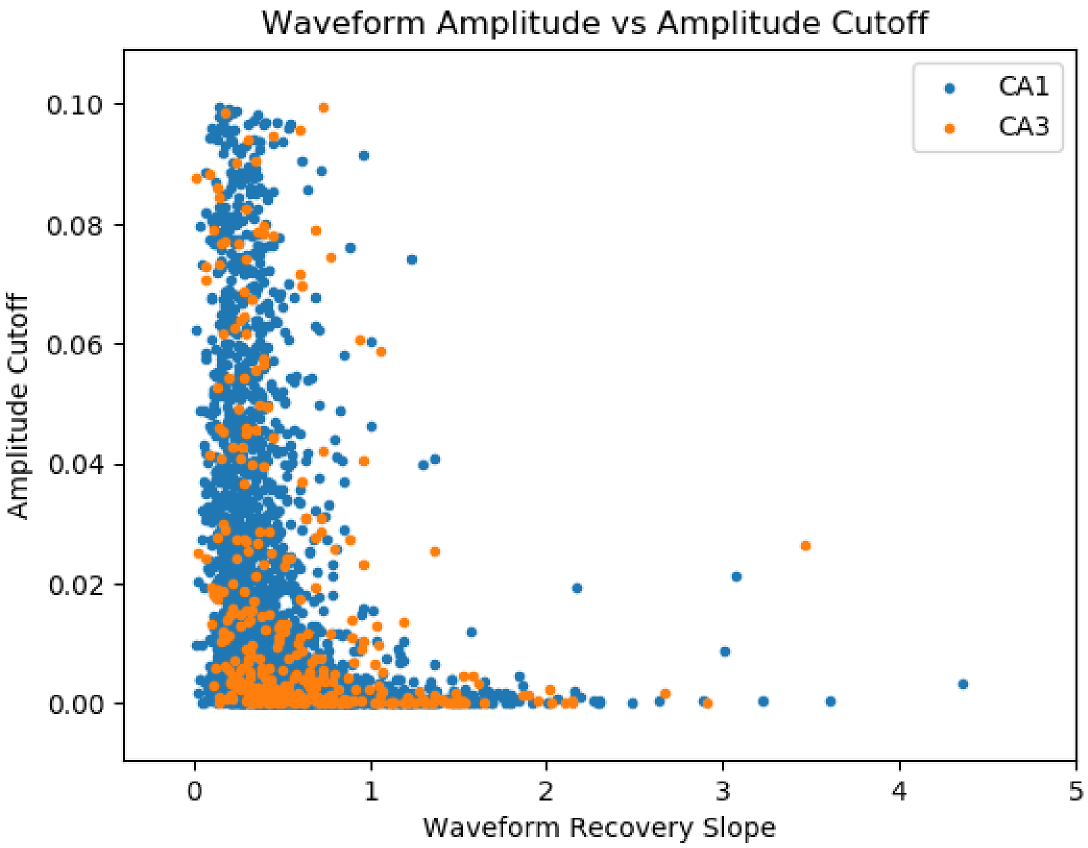
```python
CA_types = CA['ecephys_structure_acronym'].unique()
print(dendrite_types)

fig,ax = plt.subplots()

for item in dendrite_types:

    df = CA[CA['ecephys_structure_acronym'] == item]
    plt.scatter(abs(df['waveform_recovery_slope']),
                abs(df['amplitude_cutoff']),
                label=item,
               marker ='.')
ax.set_xlim(right=0.6)
plt.ylabel("Amplitude Cutoff")
plt.xlabel("Waveform Recovery Slope")
plt.legend(loc='best')
plt.title('Waveform Amplitude vs Amplitude Cutoff ')
plt.show()
```

```
['CA1' 'CA3']
```



```
[134]: CA_types = CA['ecephys_structure_acronym'].unique()
       print(dendrite_types)

       fig,ax = plt.subplots()

       for item in dendrite_types:

           df = CA[CA['ecephys_structure_acronym'] == item]
           plt.scatter(abs(df['waveform_repolarization_slope']),
                       abs(df['amplitude_cutoff']),
                       label=item,
                      marker ='.')
       ax.set_xlim(left=-0.4,right=5)
       plt.ylabel("Amplitude Cutoff")
       plt.xlabel("Waveform Recovery Slope")
       plt.legend(loc='best')
       plt.title('Waveform Amplitude vs Amplitude Cutoff ')
       plt.show()
```

```
['CA1' 'CA3']
```



10 Data Analysis & Results

Now we will interpret what our data is telling us. While the violin plots from above give us an idea of how many scores fall at a certain slope, we cannot tell from the graph whether or not the differences between CA1 and CA3 are significant.

10.0.1 The following codes will determine whether our data is skewed across all variables observed : Waveform repolarization, firing rate, and waveform amplitude.

Based on the results from the eigenvectors, PCA, and kmeans, we followed up by exploring possible relationships between variables that were observed through out Neuropixels.

We can start by extracting data from Allen Brain Observatory and storing in manifest_path. Here we will extact sessions to give us a broad overview of our data.

```
[48]: #Checks if CA1 data is skewed; significant p-values indicate data is skewed; yes
      stats.skewtest(CA1['waveform_repolarization_slope'])
```

```
[48]: SkewtestResult(statistic=34.74001907308188, pvalue=1.9610722852060116e-264)
```

```
[49]: #Checks if CA1 data is skewed; yes.
      stats.skewtest(CA3['waveform_repolarization_slope'])
```

```
[49]: SkewtestResult(statistic=11.738146238345339, pvalue=8.124779869250301e-32)
```

```
[50]: #Checks if CA1 data is skewed; yes
      stats.skewtest(CA1['firing_rate'])
```

```
[50]: SkewtestResult(statistic=39.50659925463708, pvalue=0.0)
```

```
[51]: #Checks if CA3 data is skewed; yes
      stats.skewtest(CA3['firing_rate'])
```

```
[51]: SkewtestResult(statistic=11.711844202092166, pvalue=1.1084210545909538e-31)
```

```
[52]: #Checks if CA1 data is skewed; yes.
      stats.skewtest(CA1['waveform_amplitude'])
```

```
[52]: SkewtestResult(statistic=32.31748640466235, pvalue=3.9735803456291886e-229)
```

```
[53]: #Checks if CA3 data is skewed; yes.
      stats.skewtest(CA3['waveform_amplitude'])
```

```
[53]: SkewtestResult(statistic=11.422913111454644, pvalue=3.212809965956617e-30)
```

Based on our Skewtests, we can see that all of our data was indeed, skewed.

10.0.2 The following will report the median for CA1 and CA3, across our 3 conditions

```
[54]: #Report median instead of mean if skewed.
      CA1_WRS_median=np.median(CA1['waveform_repolarization_slope'])
      print(CA1_WRS_median)
```

```
0.4011383847297325
```

```
[55]: #Report median instead of mean if skewed.
      CA3_WRS_median=np.median(CA3['waveform_repolarization_slope'])
      print(CA3_WRS_median)
```

```
0.45600688318860655
```

```
[56]: #Report median instead of mean if skewed.
      CA1_FR_median=np.median(CA1['firing_rate'])
      print(CA1_FR_median)
```

```
4.7049727363730245
```

```
[57]: #Report median instead of mean if skewed.
      CA3_FR_median=np.median(CA3['firing_rate'])
      print(CA3_FR_median)
```

```
6.114173644915274
```

```
[58]: #Report median instead of mean if skewed.
      CA1_WA_median=np.median(CA1['waveform_amplitude'])
      print(CA1_WA_median)
```

```
131.26229999999998
```

```
[59]: #Report median instead of mean if skewed.
      CA3_WA_median=np.median(CA3['waveform_amplitude'])
      print(CA3_WA_median)
```

```
143.98683
```

10.0.3 We will now utilize the Bonferroni Correction which will yield our new p-value.

```
[60]: #Number of planned comparisons: 3
      bonferroni = 0.05/3
      print("New p-value: ",bonferroni)
```

```
New p-value:  0.016666666666666666
```

We will use this new p-value to determine whether or not the values observed between CA1 and CA3 are statistically significant

10.0.4 Here we will use Mann Whitney Yu statistical tests

Since our data is all skewed, and we have 2 groups in our experiment, we will be using the Mann Whitney Yu statistical test. Note that although CA1 and CA3 interact with each other and other brain regions, due to the limitation in our resources, we treated these two regions as independent.

[61]:
```
#Independent skewed samples using Mann Whitney
stats.
 ↪mannwhitneyu(CA1['waveform_repolarization_slope'],CA3['waveform_repolarization_slope'])
```

[61]:
```
MannwhitneyuResult(statistic=591591.0, pvalue=6.528642835409541e-05)
```

[62]:
```
#Independent skewed samples using Mann Whitney
stats.mannwhitneyu(CA1['firing_rate'],CA3['firing_rate'])
```

[62]:
```
MannwhitneyuResult(statistic=564318.0, pvalue=1.0692748995730874e-07)
```

[63]:
```
#Independent skewed samples using Mann Whitney
stats.mannwhitneyu(CA1['waveform_amplitude'],CA3['waveform_amplitude'])
```

[63]:
```
MannwhitneyuResult(statistic=606580.0, pvalue=0.0010450946183217442)
```

As you can see, all our tests were significant! This suggests that there is a difference regarding waveform amplitude, firing rates, and the waveform repolarization slopes between CA1 and CA3 regions!

10.1 Exploratory Analyses

We will know execute statistical tests to determine if there is a relationship between waveform recovery and amplitude cutoff, and amplitude cutoff and repolarization.

[64]:
```
##Determines correlation in CA1
CA1 = CA1.dropna()
CA1_RS = (abs(CA1['waveform_recovery_slope']).to_numpy())
CA1_AC = (abs(CA1['amplitude_cutoff'])).to_numpy()
r , p = sp.stats.pearsonr(CA1_RS,CA1_AC)
print("Pearson's r:",r,
      "p-value", p)
```

```
Pearson's r: -0.17865642122350825 p-value 6.793626918274198e-23
```

[65]:
```
##Determines correlation in CA3
CA3 = CA3.dropna()
CA3_RS = (abs(CA3['waveform_recovery_slope']).to_numpy())
CA3_AC = (abs(CA3['amplitude_cutoff'])).to_numpy()
r,p = sp.stats.pearsonr(CA3_RS,CA3_AC)
print("Pearson's r:",r,
      "p-value", p)
```

```
Pearson's r: -0.22994980490804093 p-value 6.651328531173284e-06
```

Although our data was not linear and slightly weak, there is still a relationship between the amplitude cutoff an recovery slope. As the recovery slope increases, the amplitude cutoff decreases. It appears that CA3 has a stronger correlation.

[66]:

```
##Determines correlation in CA1
CA1_Repol = (abs(CA1['waveform_repolarization_slope']).to_numpy())
CA1_AC = (abs(CA1['amplitude_cutoff'])).to_numpy()
r , p = sp.stats.pearsonr(CA1_Repol,CA1_AC)
print("Pearson's r:",r,
      "p-value", p)
```

```
Pearson's r: -0.33294207601707104 p-value 2.0264519391766072e-78
```

[67]:

```
##Determines correlation in CA3
CA3_Repol = (abs(CA3['waveform_repolarization_slope']).to_numpy())
CA3_AC = (abs(CA3['amplitude_cutoff'])).to_numpy()
r , p  = sp.stats.pearsonr(CA3_Repol,CA3_AC)
print("Pearson's r:",r,
      "p-value", p)
```

```
Pearson's r: -0.2927929555093852 p-value 7.214910422184291e-09
```

After applying Bonferroni's correction, significance is still attained.

11 Conclusion

In order to investigate some electrophysiological properties between CA1 and CA3 pyramidal cells within the hippocampus we gatheed information from the literature, andimiat utilized key terms that we believed may be expressed more in one region over the other. For example, LTP was more commonly discussed in CA1 than CA3. Furthermore, pyramidal cells was more highly mentioned in CA1 over CA3 as well. This suggests that there are indeed morphological and electrophysiological differences. We also compared the CA regions to the hippocampus because we wanted to observe how unique the CA regions are. We first used NeuroElectro dataset to gather electrophysiology data on CA regions. Although it seemed promising, unfortunately for us there was a lot of missing data regardinIn order to investigate some electrophysiological properties between CA1 and CA3 pyramidal cells within the hippocampus, we gathered information from the literature and utilized key terms that we believed may be expressed more in one region over the other. For example, LTP was more commonly discussed in CA1 than CA3. Furthermore, pyramidal cells was more highly mentioned in CA1 over CA3 as well. This suggests that there are indeed morphological and electrophysiological differences. We also compared the CA regions to the hippocampus because we wanted to observe how unique the CA regions are. We first used NeuroElectro dataset to gather electrophysiology data on CA regions. Although it seemed promising, unfortunately for us there was a lot of missing data regarding CA regions. We were still able to observe age, input resistance, and resting membrane potential in both CA regions. We coded for covariance matrices and eigenvectors but were not able to find any relationships among the variables. This is highly likely due to our low sample size. We were however, able to utilize NeuroPixels, and filter out CA regions to observe other electrophysiology properties although they were not the same variables as the ones from Neuro Electro, they were still very informative. After applying the Bonferroni correction and using the Mann Whitney Yu tests, we found that firing rate ($p<0.017$), waveform repolarization slope ($p<0.017$), and waveform amplitude ($p<0.017$) were all statically different between CA1 and CA3 regions. We also found that they were all greatly skewed (all; $p<0.05$) with CA3 (repolarization=0.46, firing rate=6.11, amplitude=143.99) having higher median values in all three variables comapred to CA1 (repolarization=0.40, firing rate=4.70, amplitude=131.26). From our eigenvectors in NeuroPixels we determined that there could possibly be a relationship between amplitude cutoff and Waveform recovery slope, or waveform repolarization slope. After executing Pearson's r test, we found that there is a slight negative relationship in waveform recovery slope and amplitude cutoff in CA1 ($r=-0.18$; $p<0.017$) and CA3 ($r=-0.23$; $p<0.017$). We also found a slight negative relationship between waveform repolarization slope and amplitude cutoff in CA1 ($r=-0.33$; $p<0.017$) and CA3 ($r=-0.29$; $p<0.017$).g CA regions. We were still able to observe age, input resistance, and resting membrane potential in both CA regions. We coded for covariance matrices and eigenvectors but were not able to find any relationships among the variables. This is highly likely due to our low sample size. We were however, able to utilize NeuroPixels, and filter out CA regions to observe other electrophysiology properties although they were not the same variables as the ones from Neuro Electro, they were still very informative. After applying the Bonferroni correction and using the Mann Whitney Yu tests, we found that firing rate ($p<0.017$), waveform repolarization slope ($p<0.017$), and waveform amplitude ($p<0.017$) were all statically different between CA1 and CA3 regions. We also found that they were all positively skewed (all; $p<0.05$). From our eigenvectors in NeuroPixels we determined that there could possibly be a relationship between amplitude cutoff and Waveform recovery slope, or waveform repolarization slope. After executing Pearson's r test, we found that there is a slight negative relationship in waveform recovery slope and amplitude cutoff in CA1 ($r=-0.18$; $p<0.017$) and CA3 ($r=-0.23$; $p<0.017$). We also found a slight negative relationship between waveform repolarization slope and amplitude cutoff in CA1 ($r=-0.33$; $p<0.017$) and CA3 ($r=-0.29$; $p<0.017$).

11.0.1 Limitations

NeuroElectro has so much potential to offer various other eletrophysiological properties that are not otherwise present in NeuroPixels. They also specifically showed different neurons and cells within CA1 such as pyramidal cells. However, their observations for CA altogether, were insufficient to conduct statistical testing. While NeuroPixels made up for the lack of samples, it did not offer specific cells like NeuroElectro did. For that reason, we cannot assume that our findings are restricted to CA pyramidal neurons, but rather CA cells on average. There are studies indicating that even within CA cells, are differences. For future directions, we hope within the next years NeuroElectro updates their databases in order to draw more data and make more accurate experimental sessions. We also would like to follow up on our previous findings on relationships between amplitude cutoff and repolarization or recovery and look through other databases to see if we obtain similar results. If so, then this relationship is crucial for understanding how LTP function and how neural networks are influenced.

12 Reflection

Working on this project has been quite the experience as all the members of our group were either beginners at python or completely new at coding. At first, we were excited about exploring neuroscience from the perspective of data science but quickly found it challenging coming up with a hypothesis or two distinct cell types to compare. We then decided to do our project on the iconic CA3 and CA1 hippocampal neurons since there seems to be a ton of information on them in the literature. We realized that publicly available databases were hard to get data mainly because they were all structured differently, required ambiguous code that didn't make much sense to extract the data, or did not contain information useful for our project. Running into these issues required a lot of time to work around them and find data worth using. By the time we completed the first project, we had a greater insight as to what and were to look for more data which made it more intuitive to work on the final project.