

Final Project

April 7, 2020

1 The differences in electrophysiology, morphology and gene expression in mice visual cortex.

1.1 Team Members

- Martyna Marcinkowska: Background, discussion, editing, PCA.
- Justin Lee: Analysis and visualisation of the electrophysiology data and morphology data.
- Khoa Huynh: Analysis and visualisation of the ISH data and morphology data.

1.2 Abstract

We compared electrophysiology and morphology data from the Allen Cell Type dataset and gene expression data from the Allen In Situ Hybridization (ISH) dataset between layer 2/3 and 5 of the mouse primary visual cortex. We found that the layers' electrophysiology differed significantly for rheobase, adaptation index, membrane time constant, ramp spike time and the resting potentials which is in line with other findings while the morphology differed significantly for max Euclidean distance only for spiny neurons and all of max Euclidean distance, number of stems, number of bifurcations, average contractions, and parent:daughter ratios for aspiny neurons. We also found at least 50% fold change gene expression between almost 50% of genes included in the analysis and the differences between specific genes in the layers could at least partly explain the electrophysiological differences.

1.3 Research Question

Are the differences in electrophysiological, morphological features and fold change gene expression between mice visual primary cortex layer 2/3 and 5 linked?

2 Background and Prior Work

The neocortex has a characteristic laminar organization where each layer has both excitatory and inhibitory neurons. Excitatory neurons, which constitute 80-85% of all cortical neurons, are primarily pyramidal neurons while inhibitory are more diverse in morphology and electrophysiological properties (Beaulieu, 1993). Through investigation of each layer, their unique inputs, projection

targets and functions have been found. The differences or similarities between them might be a result of underlying differences in neuron classes and their electrophysiological properties.

Mouse visual cortex consists of 6 physiological layers and further sublayers. Their properties e.g. firing rates, burstiness etc. display unique layer and brain state dependence (Senzai et al., 2019). Previous research has shown that in rodents, there is no clear boundary between layer 2 (L2) and L3 of the cortex which are referred to as L2/3. L2/3 excitatory neurons integrate information across cortical areas and hemispheres. It has been also found that pyramidal neurons are the main neuron type in this layer. In sensory cortices, different L2/3 neurons have been shown to have different sound-responsiveness or visual selectivity (Luo et al., 2017). Layer 5 neurons primarily connect the cerebral cortex with subcortical regions. Similarly, this layer is prevalently made up of pyramidal neurons. It is best developed in motor cortical areas (Swenson, 2006).

In order to analyse the differences in electrophysiology, morphology and gene expression, we will be using the Allen ISH and Allen Cell Types datasets. Both datasets are part of the Allen Brain Atlas which is a publicly available online resource of information about the human and mouse brain. The *in situ* hybridization dataset comprises multiple projects' results using an 'all genes, all structures' survey approach that generated genome-wide gene expression profiles in mouse brains. Mouse brain slices were analysed using automated high-throughput procedures alongside anatomic and histologic data to detect and label RNA sequences in different sections of brain tissue, ultimately forming an atlas of approximately 20,000 distinct mouse genes. Similarly, the brain cell type database is a survey of biological features measured from single cells in human and mouse. For the mouse data, cells are acquired from selected brain areas in the adult mouse and identified for isolation using transgenic mouse lines with fluorescent reporters: excitatory cells with layer-enriched distribution and inhibitory cells expressing canonical markers were isolated for analysis of electrophysiology and morphology. In the electrophysiology and morphology datasets, there are 2331 cells analysed. The electrophysiology data was collected using whole cell patch clamp recordings performed with a range of different stimuli. The morphology data was produced by filling the cells with biocytin and examining fluorescent and brightfield images obtained at whole-section or single cell resolutions. Serial images obtained at 63X were obtained for morphological analysis and images at 20X magnification were used to determine positioning and anatomical location of each neuron. We'll focus on the primary visual cortex which is well-described and shares the characteristic laminar organisation.

The differences between the layers of the visual cortex of mice are still not fully understood and their examination may provide an insight into the differences seen in humans and the functions of these different layers. The aims of this proposal are to further examine electrophysiological, morphological and transcriptomic differences between the different visual cortex layers and compare them with previously established findings.

2.0.1 References:

- (1) Tasic B, Yao Z, Graybiel LT, et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*. 2018;563(7729):72–78. doi:10.1038/s41586-018-0654-5

Accessed through: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6456269/>

- (2) Senzai Y, Fernandez-Ruiz A, Buzsáki G. Layer-Specific Physiological Features and Interlaminar Interactions in the Primary Visual Cortex of the Mouse. *Neuron*. 2019; 101(3): 500–513.

doi:10.1016/j.neuron.2018.12.009

Accessed through: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6367010/>

- (3) Belgard T, Marques A, Oliver P, et al. A Transcriptomic Atlas of Mouse Neocortical Layers. *Neuron*. 2011; 71(4): 605-616. doi:10.1016/j.neuron.2011.06.039.

Accessed through: <https://www.sciencedirect.com/science/article/pii/S0896627311006015>

- (4) Luo H, Hasegawa K, Liu M, Song WJ. Comparison of the Upper Marginal Neurons of Cortical Layer 2 with Layer 2/3 Pyramidal Neurons in Mouse Temporal Cortex. *Frontiers in Neuroanatomy*. 2017; 11: 115. doi:10.3389/fnana.2017.00115

Accessed through: <https://www.frontiersin.org/articles/10.3389/fnana.2017.00115/full>

- (5) Swenson R. Review of Clinical and Functional Neuroscience. Dartmouth Medical School. 2006; chapter 11.

Accessed through: https://www.dartmouth.edu/~rswenson/NeuroSci/chapter_11.html

- (6) Watakabe A, Hirokawa J, Ichinohe N, Ohsawa S, Kaneko T, Rockland KS, Yamamori T. Area-specific stratification of deep layer neurons in the rat cortex. *The Journal of Comparative Neurology*. 2012; 520(16): 3553-3573. doi:10.1002/cne.23160

Accessed through: <https://onlinelibrary.wiley.com/doi/full/10.1002/cne.23160>

- (7) Beaulieu C. Numerical data on neocortical neurons in adult rat, with special reference to the GABA population. *Brain Research*. 1993; 609(1-2): 284-292. doi:10.1016/0006-8993(93)90884-P

Accessed through: <http://www.sciencedirect.com/science/article/pii/000689939390884P>

- (8) Scheuss, V., & Bonhoeffer, T. (2013). Function of Dendritic Spines on Hippocampal Inhibitory Neurons. *Cerebral Cortex*, 24(12), 3142-3153. doi: 10.1093/cercor/bht171

Accessed through: <https://academic.oup.com/cercor/article/24/12/3142/273136>

- (9) Bopp, R., Costa, N. M. D., Kampa, B. M., Martin, K. A. C., & Roth, M. M. (2014). Pyramidal Cells Make Specific Connections onto Smooth (GABAergic) Neurons in Mouse Visual Cortex. *PLoS Biology*, 12(8). doi: 10.1371/journal.pbio.1001932

Accessed through: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4138028/>

2.1 Hypothesis

The differences between electrophysiological and morphological features and gene expression fold change between layer 2/3 and layer 5 in mouse primary visual cortex are correlated.

3 Data Analysis

3.1 Setup: Electrophysiology

In this section, we import the appropriate packages that will be needed to both plot and analyze the electrophysiology data. We will also import the CellTypesCache module to be able to extract electrophysiology data from the Cell types data base.

3.2 Data Wrangling: Electrophysiology

```
[42]: #Code to import matplotlib and pyplot module
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

#Code to import the numpy toolbox
import numpy as np

#Code to import pandas as pd
import pandas as pd

#Code to import scipy as stats
from scipy import stats

# Code to import the needed toolboxes from the Allensdk
from allensdk.core.cell_types_cache import CellTypesCache
from allensdk.api.queries.cell_types_api import CellTypesApi
```

In this section we extract the mouse metadata and the electrophysiology data into dataframes and then join the two dataframes together.

```
[43]: #Initialize the cache as ctc
ctc = CellTypesCache(manifest_file='cell_types/manifest.json')

# Turn mouse metadata into dataframe and set the index to be ID
mouse_df = pd.DataFrame(ctc.get_cells(species=[CellTypesApi.MOUSE])).
    ↪set_index('id')

# Turn ephys data into a dataframe and set the index to be specimen ID
ephys_features = pd.DataFrame(ctc.get_ephys_features()).set_index('specimen_id')

# Join the two dataframes together
mouse_ephys_df = mouse_df.join(ephys_features)
```

3.3 Data Analysis & Results: Electrophysiology

We want to compare the two layers by plotting 9 boxplots contrasting the rheobase (minimal current amplitude resulting in a depolarization threshold being reached), adaptation indices, frequency-current (F-I) curve slopes, input resistances, membrane time constraints, ramp spike times, upstroke-downstroke ratios, average interspike interval (ISI), and resting membrane potentials.

```
[44]: # Set up a figure with 9 subplots
fig,ax = plt.subplots(3,3,figsize=(20,20))

mouse_df = mouse_ephys_df[mouse_ephys_df['structure_area_abbrev']=='VISp']

mouse_2_3_df = mouse_df[mouse_df['structure_layer_name']=='2/3']
mouse_5_df = mouse_df[mouse_df['structure_layer_name']=='5']

#On the first graph, create the box plot of Adaptation Index
adaptation_mouse_2_3_ft = mouse_2_3_df['adaptation'].dropna()
adaptation_mouse_5_ft = mouse_5_df['adaptation'].dropna()

adaptation_data = [adaptation_mouse_2_3_ft,adaptation_mouse_5_ft]

ax[0,0].boxplot(adaptation_data)
ax[0,0].set_xticklabels(['Layer 2/3','Layer 5'])
ax[0,0].set_ylabel('Adaptation Index')
ax[0,0].set_title('Boxplot of Adaption Index')

#On the second graph, create the box plot of Rheobase

threshold_i_long_square_mouse_2_3_ft = mouse_2_3_df['threshold_i_long_square']
threshold_i_long_square_mouse_5_ft = mouse_5_df['threshold_i_long_square']

threshold_i_long_square_data = □
    ↳ [threshold_i_long_square_mouse_2_3_ft,threshold_i_long_square_mouse_5_ft]

ax[0,1].boxplot(threshold_i_long_square_data)
ax[0,1].set_xticklabels(['Layer 2/3','Layer 5'])
ax[0,1].set_ylabel('Rheobase (pA)')
ax[0,1].set_title('Boxplot of Rheobase')

#On the third graph, create the box plot of F-I curve slopes
f_i_curve_slope_mouse_2_3_ft = mouse_2_3_df['f_i_curve_slope']
f_i_curve_slope_mouse_5_ft = mouse_5_df['f_i_curve_slope']

f_i_curve_slope_data = [f_i_curve_slope_mouse_2_3_ft,f_i_curve_slope_mouse_5_ft]
```

```

ax[0,2].boxplot(f_i_curve_slope_data)
ax[0,2].set_xticklabels(['Layer 2/3','Layer 5'])
ax[0,2].set_ylabel('F-I curve slopes (spike/s/pA)')
ax[0,2].set_title('Boxplot of F-I curve slopes')

#On the fourth graph, create a box plot of Input Resistance

input_resistance_mohm_mouse_2_3_ft = mouse_2_3_df['input_resistance_mohm'].
↳dropna()
input_resistance_mohm_mouse_5_ft = mouse_5_df['input_resistance_mohm'].dropna()

input_resistance_mohm_data = ␣
↳[input_resistance_mohm_mouse_2_3_ft,input_resistance_mohm_mouse_5_ft]

ax[1,0].boxplot(input_resistance_mohm_data)
ax[1,0].set_xticklabels(['Layer 2/3','Layer 5'])
ax[1,0].set_ylabel('Input Resistance (mΩ)')
ax[1,0].set_title('Boxplot of Input Resistance')

#On the fifth graph, create a box plot of Membrane Time Constant

tau_mouse_2_3_ft = mouse_2_3_df['tau'].dropna()
tau_mouse_5_ft = mouse_5_df['tau'].dropna()

tau_data = [tau_mouse_2_3_ft,tau_mouse_5_ft]

ax[1,1].boxplot(tau_data)
ax[1,1].set_xticklabels(['Layer 2/3','Layer 5'])
ax[1,1].set_ylabel('Membrane Time Constant (ms)')
ax[1,1].set_title('Boxplot of Membrane Time Constant')

#On the sixth graph, create a box plot of ramp spike times

peak_t_ramp_mouse_2_3_ft = mouse_2_3_df['peak_t_ramp'].dropna()
peak_t_ramp_mouse_5_ft = mouse_5_df['peak_t_ramp'].dropna()

peak_t_ramp_data = [peak_t_ramp_mouse_2_3_ft,peak_t_ramp_mouse_5_ft]

ax[1,2].boxplot(peak_t_ramp_data)
ax[1,2].set_xticklabels(['Layer 2/3','Layer 5'])
ax[1,2].set_ylabel('Ramp Spike Time (s)')
ax[1,2].set_title('Boxplot of Ramp Spike Time')

# On the seventh graph, create the box plot of upstroke-downstroke ratio
upstroke_downstroke_ratio_long_square_mouse_2_3_ft = ␣
↳mouse_2_3_df['upstroke_downstroke_ratio_long_square'].dropna()

```

```

upstroke_downstroke_ratio_long_square_mouse_5_ft =
↳mouse_5_df['upstroke_downstroke_ratio_long_square'].dropna()

upstroke_downstroke_ratio_long_square_data =
↳[upstroke_downstroke_ratio_long_square_mouse_2_3_ft,upstroke_downstroke_ratio_long_square_m

ax[2,0].boxplot(upstroke_downstroke_ratio_long_square_data)
ax[2,0].set_xticklabels(['Layer 2/3','Layer 5'])
ax[2,0].set_ylabel('Upstroke-Downstroke Ratio')
ax[2,0].set_title('Boxplot of Upstroke-Downstroke Ratio')

# On the eight graph, create the box plot of average ISI
avg_isi_mouse_2_3_ft = mouse_2_3_df['avg_isi'].dropna()
avg_isi_mouse_5_ft = mouse_5_df['avg_isi'].dropna()

avg_isi_data = [avg_isi_mouse_2_3_ft,avg_isi_mouse_5_ft]

ax[2,1].boxplot(avg_isi_data)
ax[2,1].set_xticklabels(['Layer 2/3','Layer 5'])
ax[2,1].set_ylabel('Average ISI (ms)')
ax[2,1].set_title('Boxplot of Average ISI')

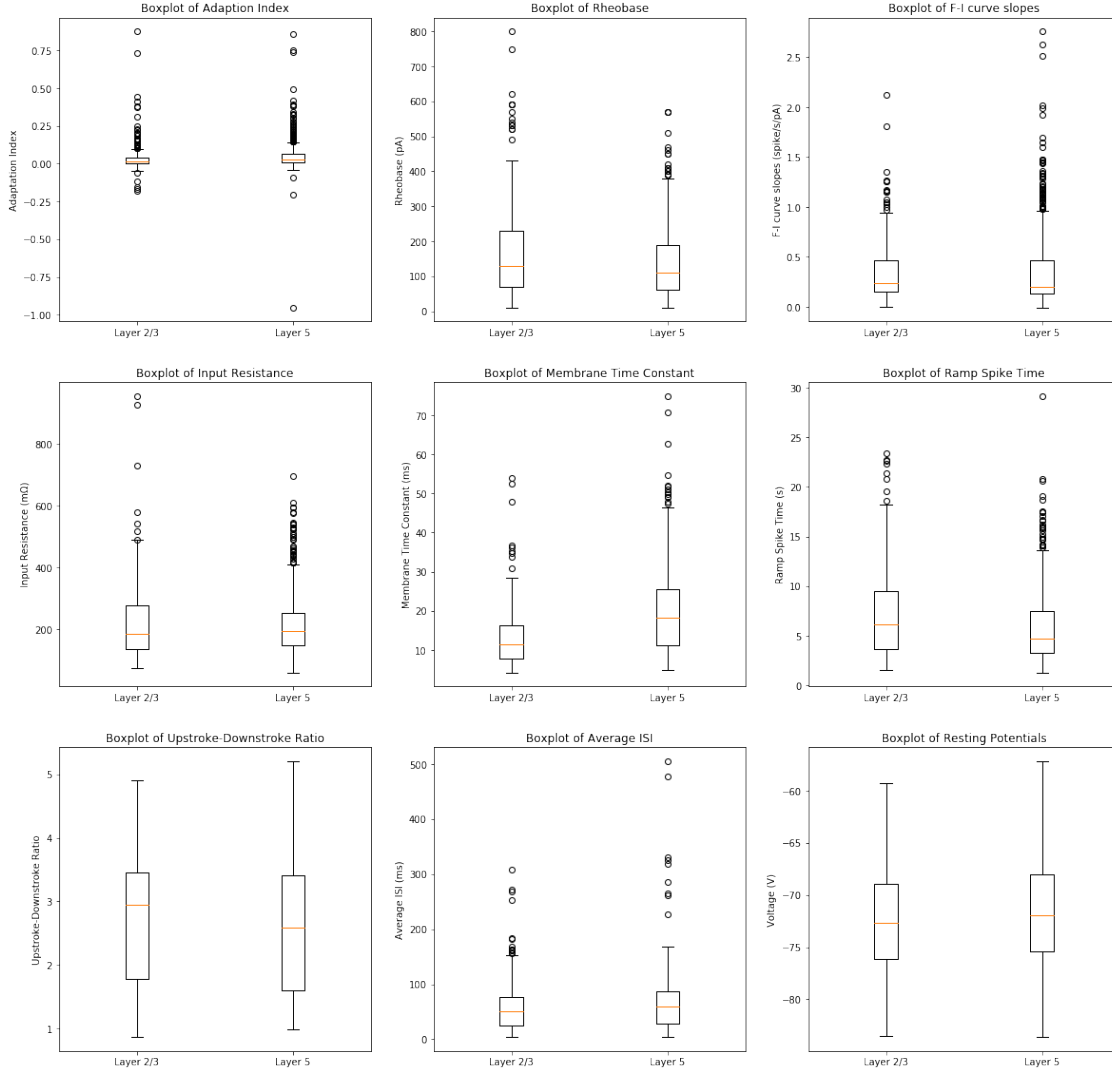
# On the ninth graph, create the box plot of resting potential
vrest_mouse_2_3_ft = mouse_2_3_df['vrest'].dropna()
vrest_mouse_5_ft = mouse_5_df['vrest'].dropna()

vrest_data = [vrest_mouse_2_3_ft,vrest_mouse_5_ft]

ax[2,2].boxplot(vrest_data)
ax[2,2].set_xticklabels(['Layer 2/3','Layer 5'])
ax[2,2].set_ylabel('Voltage (V)')
ax[2,2].set_title('Boxplot of Resting Potentials')

plt.show()

```



We wanted to statistically compare the various electrophysiological differences between the two layers. For each different physiological feature, a skew test was done to determine if the data represented a normal distribution or a skewed distribution. If it was a normal distribution, a t-test was performed. If there was a skewed distribution, a Mann Whitney test was performed. To correct for multiple comparisons, the Bonferroni correction was used. Accordingly, any p-values above 0.00556 are not considered significant.

```
[45]: #Statistical Analysis of Adaptation Index
adaptation_mouse_2_3_skewed_stats, adaptation_mouse_2_3_skewed = stats.
↳skewtest(adaptation_mouse_2_3_ft)
adaptation_mouse_5_skewed_stats, adaptation_mouse_5_skewed = stats.
↳skewtest(adaptation_mouse_5_ft)
print('For Adaptation Index:')
if adaptation_mouse_2_3_skewed > 0.05 and adaptation_mouse_5_skewed > 0.05:
```



```

    print('Normal Distributions. The T-test reports:')
    print(stats.ttest_ind(adaptation_mouse_2_3_ft,adaptation_mouse_5_ft))
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.mannwhitneyu(adaptation_mouse_2_3_ft,adaptation_mouse_5_ft))
    print()

#Statistical Analysis of Rheobase
threshold_i_long_square_mouse_2_3_skewed_stats,␣
    ↳threshold_i_long_square_mouse_2_3_skewed = stats.
    ↳skewtest(threshold_i_long_square_mouse_2_3_ft)
threshold_i_long_square_mouse_5_skewed_stats,␣
    ↳threshold_i_long_square_mouse_5_skewed = stats.
    ↳skewtest(threshold_i_long_square_mouse_5_ft)
print('For Rheobase:')
if threshold_i_long_square_mouse_2_3_skewed > 0.05 and␣
    ↳threshold_i_long_square_mouse_5_skewed > 0.05:
    print('Normal Distributions. The T-test reports:')
    print(stats.
    ↳ttest_ind(threshold_i_long_square_mouse_2_3_ft,threshold_i_long_square_mouse_5_ft))
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.
    ↳mannwhitneyu(threshold_i_long_square_mouse_2_3_ft,threshold_i_long_square_mouse_5_ft))
    print()

#Statistical Analysis of F-I Slopes
f_i_curve_slope_mouse_2_3_skewed_stats, f_i_curve_slope_mouse_2_3_skewed =␣
    ↳stats.skewtest(f_i_curve_slope_mouse_2_3_ft)
f_i_curve_slope_mouse_5_skewed_stats, f_i_curve_slope_mouse_5_skewed = stats.
    ↳skewtest(f_i_curve_slope_mouse_5_ft)
print('For F-I Slopes:')
if f_i_curve_slope_mouse_2_3_skewed > 0.05 and f_i_curve_slope_mouse_5_skewed >␣
    ↳0.05:
    print('Normal Distributions. The T-test reports:')
    print(stats.
    ↳ttest_ind(f_i_curve_slope_mouse_2_3_ft,f_i_curve_slope_mouse_5_ft))
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.
    ↳mannwhitneyu(f_i_curve_slope_mouse_2_3_ft,f_i_curve_slope_mouse_5_ft))
    print()

```

```

#Statistical Analysis of Input Resistance
input_resistance_mohm_mouse_2_3_skewed_stats,␣
    ↳input_resistance_mohm_mouse_2_3_skewed = stats.
    ↳skewtest(input_resistance_mohm_mouse_2_3_ft)
input_resistance_mohm_mouse_5_skewed_stats,␣
    ↳input_resistance_mohm_mouse_5_skewed = stats.
    ↳skewtest(input_resistance_mohm_mouse_5_ft)
print('For Input Resistance:')
if input_resistance_mohm_mouse_2_3_skewed > 0.05 and␣
    ↳input_resistance_mohm_mouse_5_skewed > 0.05:
    print('Normal Distributions. The T-test reports:')
    print(stats.
    ↳ttest_ind(input_resistance_mohm_mouse_2_3_ft,input_resistance_mohm_mouse_5_ft))
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.
    ↳mannwhitneyu(input_resistance_mohm_mouse_2_3_ft,input_resistance_mohm_mouse_5_ft))
    print()

#Statistical Analysis of Membrane Time Constant
tau_mouse_2_3_skewed_stats, tau_mouse_2_3_skewed = stats.
    ↳skewtest(tau_mouse_2_3_ft)
tau_mouse_5_skewed_stats, tau_mouse_5_skewed = stats.skewtest(tau_mouse_5_ft)
print('For Membrane Time Constant:')
if tau_mouse_2_3_skewed > 0.05 and tau_mouse_5_skewed > 0.05:
    print('Normal Distributions. The T-test reports:')
    print(stats.ttest_ind(tau_mouse_2_3_ft,tau_mouse_5_ft))
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.mannwhitneyu(tau_mouse_2_3_ft,tau_mouse_5_ft))
    print()

#Statistical Analysis of Ramp Spike Time
peak_t_ramp_mouse_2_3_skewed_stats, peak_t_ramp_mouse_2_3_skewed = stats.
    ↳skewtest(peak_t_ramp_mouse_2_3_ft)
peak_t_ramp_mouse_5_skewed_stats, peak_t_ramp_mouse_5_skewed = stats.
    ↳skewtest(peak_t_ramp_mouse_5_ft)
print('For Ramp Spike Time:')
if peak_t_ramp_mouse_2_3_skewed > 0.05 and peak_t_ramp_mouse_5_skewed > 0.05:
    print('Normal Distributions. The T-test reports:')
    print(stats.ttest_ind(peak_t_ramp_mouse_2_3_ft,peak_t_ramp_mouse_5_ft))
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')

```

```

print(stats.mannwhitneyu(peak_t_ramp_mouse_2_3_ft,peak_t_ramp_mouse_5_ft))
print()

#Statistical Analysis of Upstroke-Downstroke ratio
upstroke_downstroke_ratio_long_square_mouse_2_3_skewed_stats,
↳upstroke_downstroke_ratio_long_square_mouse_2_3_skewed = stats.
↳skewtest(upstroke_downstroke_ratio_long_square_mouse_2_3_ft)
upstroke_downstroke_ratio_long_square_mouse_5_skewed_stats,
↳upstroke_downstroke_ratio_long_square_mouse_5_skewed = stats.
↳skewtest(upstroke_downstroke_ratio_long_square_mouse_5_ft)
print('For Upstroke-Downstroke:')
if upstroke_downstroke_ratio_long_square_mouse_2_3_skewed > 0.05 and
↳upstroke_downstroke_ratio_long_square_mouse_5_skewed > 0.05:
    print('Normal Distributions. The T-test reports:')
    print(stats.
↳ttest_ind(upstroke_downstroke_ratio_long_square_mouse_2_3_ft,upstroke_downstroke_ratio_long
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.
↳mannwhitneyu(upstroke_downstroke_ratio_long_square_mouse_2_3_ft,upstroke_downstroke_ratio_l
    print()

#Statistical Analysis of Average ISI
avg_isi_mouse_2_3_skewed_stats, avg_isi_mouse_2_3_skewed = stats.
↳skewtest(avg_isi_mouse_2_3_ft)
avg_isi_mouse_5_skewed_stats, avg_isi_mouse_5_skewed = stats.
↳skewtest(avg_isi_mouse_5_ft)
print('For Average ISI:')
if avg_isi_mouse_2_3_skewed > 0.05 and avg_isi_mouse_5_skewed > 0.05:
    print('Normal Distributions. The T-test reports:')
    print(stats.ttest_ind(avg_isi_mouse_2_3_ft,avg_isi_mouse_5_ft))
    print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.mannwhitneyu(avg_isi_mouse_2_3_ft,avg_isi_mouse_5_ft))
    print()

#Statistical Analysis of Resting Potentials
vrest_mouse_2_3_skewed_stats, vrest_mouse_2_3_skewed = stats.
↳skewtest(vrest_mouse_2_3_ft)
vrest_mouse_5_skewed_stats, vrest_mouse_5_skewed = stats.
↳skewtest(vrest_mouse_5_ft)
print('For Resting Potentials:')
if vrest_mouse_2_3_skewed > 0.05 and vrest_mouse_5_skewed > 0.05:

```

```

print('Normal Distributions. The T-test reports:')
print(stats.ttest_ind(vrest_mouse_2_3_ft,vrest_mouse_5_ft))
print()
else:
    print('Skewed Distribution. The Mann Whitney test results are:')
    print(stats.mannwhitneyu(vrest_mouse_2_3_ft,vrest_mouse_5_ft))
    print()

```

For Adaptation Index:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=47536.0, pvalue=1.7009163479279876e-07)

For Rheobase:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=67934.0, pvalue=0.0007891471496786877)

For F-I Slopes:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=74591.0, pvalue=0.13304108090699795)

For Input Resistance:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=75842.0, pvalue=0.23320572550417956)

For Membrane Time Constant:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=47413.0, pvalue=1.691043385226706e-21)

For Ramp Spike Time:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=59629.0, pvalue=6.500888316348063e-06)

For Upstroke-Downstroke:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=71386.0, pvalue=0.018067425706462966)

For Average ISI:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=64825.0, pvalue=0.03099983693704147)

For Resting Potentials:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=69843.0, pvalue=0.005102703539269477)

We can see that, out of the 9 features considered, there is a significant difference (after Bonferroni correction: $p\text{-value} < 0.00556$) between the two layers only for rheobase, adaptation index, membrane time constant, ramp spike time and the resting potentials. The distributions for all analysed

attributes were skewed.

3.4 Dimensionality Reduction: Principal Component Analysis

We chose to conduct the Principal Component Analysis, or PCA for short, to reduce the dimensionality of our data. In other words, we wanted to process the data in a way that allows us to visualise the differences between the layers and their relatedness while retaining as much of the variance as possible. We chose to analyse only the features that were preselected for the electrophysiology and morphology analyses.

3.4.1 Electrophysiology PCA

```
[46]: from sklearn.decomposition import PCA
import seaborn as sns
```

```
[47]: #Preparing the dataframe for PCA
pc_mouse_df = pd.concat([mouse_2_3_df, mouse_5_df])

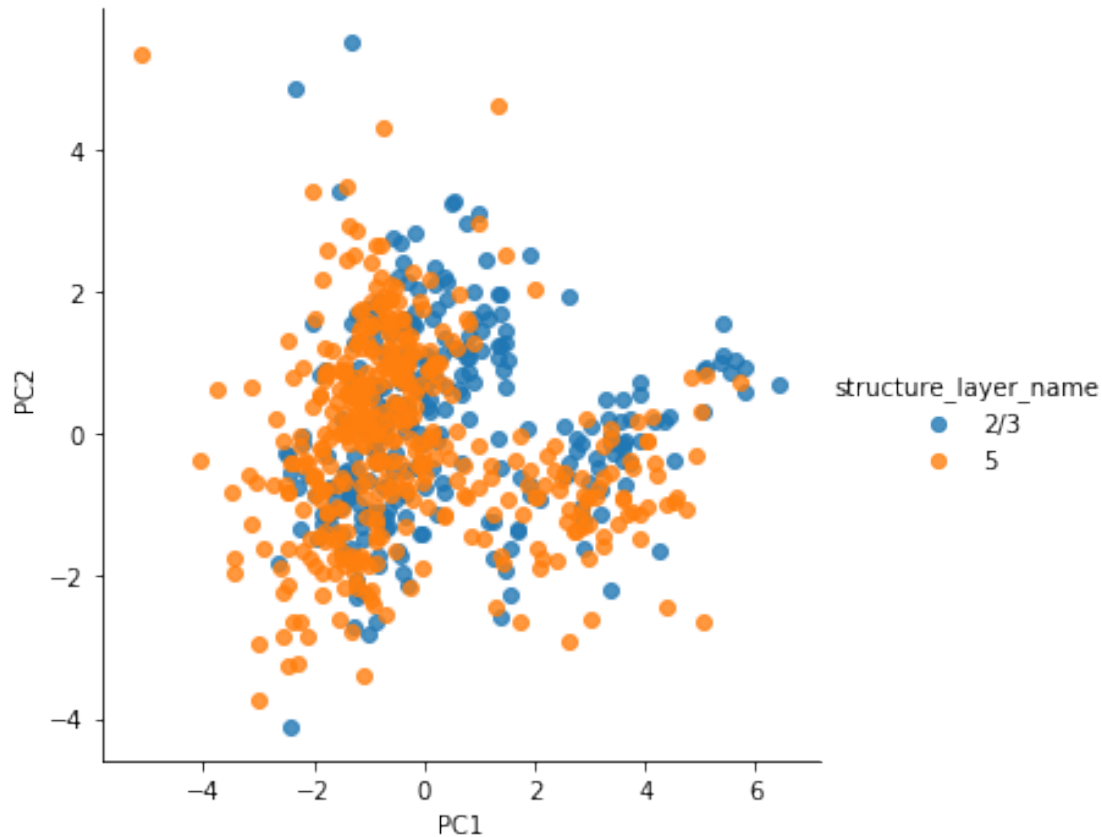
#Preparing the dataset for analysis with the 9 selected components only
x_mouse = pc_mouse_df[['adaptation', 'threshold_i_long_square',
    ↳ 'peak_t_ramp', 'vrest', 'tau', 'f_i_curve_slope', 'input_resistance_mohm', 'upstroke_downstroke_
x_mouse = x_mouse.dropna(axis=0).dropna(axis=1)

#Normalizing the data
x_mouse = (x_mouse - x_mouse.mean())/x_mouse.std()

#Preparing another dataframe with an extra layer column needed for marking plot
↳ by layer
pc_simple = x_mouse.copy()
pc_simple['structure_layer_name'] = pc_mouse_df['structure_layer_name']
```

```
[48]: #Initiating a PCA with 9 components
pca = PCA()
x_mouse_pca = pca.fit_transform(x_mouse)

#Plotting the PCA by layer
pc_simple['PC1'] = x_mouse_pca[:, 0]
pc_simple['PC2'] = x_mouse_pca[:, 1]
sns.lmplot("PC1", "PC2", hue='structure_layer_name', data=pc_simple,
    ↳ fit_reg=False)
plt.show()
```



On the plot, we see the data marked by layer. Clearly, the variance in the data is not strongly correlated with its layer. However, we see two defined clusters. We also wanted to analyse what makes up the principal components to see what are the most “significant” factors that cause the variance.

```
[49]: #Calculating the explained variance for each PC
ex_variance = np.var(x_mouse_pca,axis=0)
ex_variance_ratio = ex_variance/np.sum(ex_variance)
print(ex_variance_ratio)

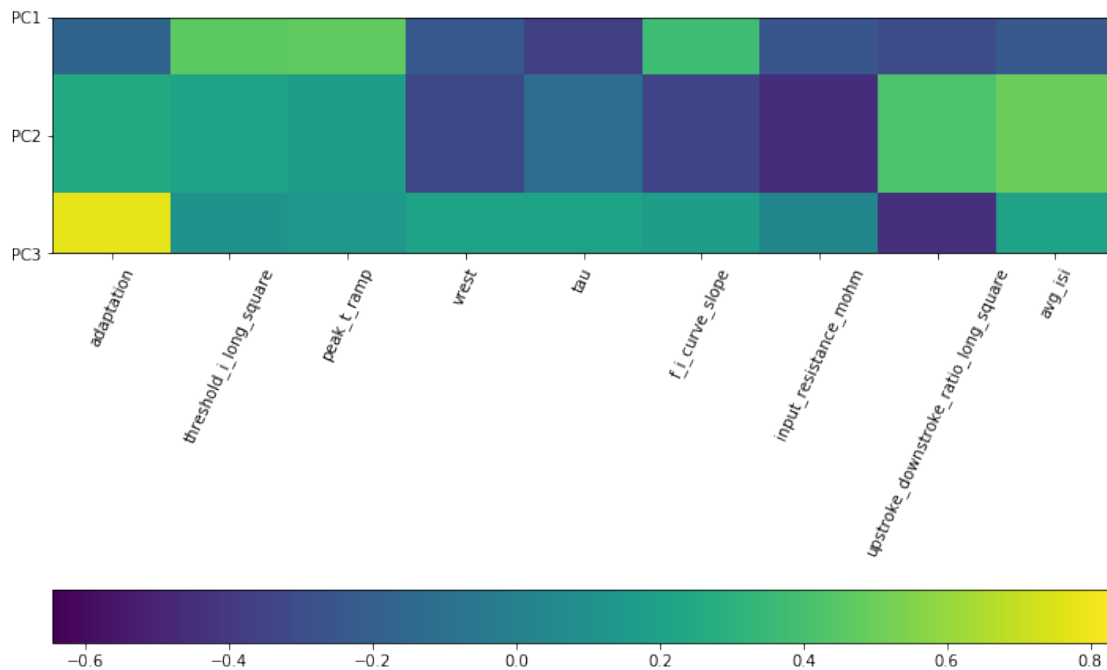
[0.40799042 0.2051752  0.11170739 0.08664379 0.06222006 0.04952887
 0.03824032 0.03103573 0.00745822]
```

The first three PCs explain over 60% of the variance.

```
[50]: #Plotting a figure explaining which features contribute to each PC
plt.figure(figsize=(10, 20))
plt.imshow(pca.components_,cmap='viridis',)
plt.yticks([0,1,2],['PC1','PC2','PC3'],fontsize=10)
plt.colorbar(orientation='horizontal')
plt.tight_layout()
```

```
plt.xticks(range(len(x_mouse.columns)),x_mouse.columns,rotation=65)

plt.show()
```



Above, we're trying to interpret what are the most meaningful factors for our dataset. We see which features contribute most to the first 3 PCs. The first PC which defines the most variance is most strongly made up of the rheobase, spike ramp time and F-I slopes.

3.5 Setup and Data Wrangling: Morphology

In this section, we extract the mouse metadata and mouse morphological data into separate dataframes and then combine them together into one.

```
[51]: # Obtain mouse metadata into a dataframe and set the index to be ID
mouse_df = pd.DataFrame(ctc.get_cells(species=[CellTypesApi.MOUSE])).
    ↪set_index('id')

# Put morphological data into a dataframe and set the index to be specimen ID
morphology_features = pd.DataFrame(ctc.get_morphology_features()).
    ↪set_index('specimen_id')

# Join the two dataframes together
mouse_morph_df = mouse_df.join(morphology_features)

# Show the first 5 rows
```

```
mouse_morph_df.head()
```

```
[51]:
```

	reporter_status	cell_soma_location \
313860745	positive	[9582.4439484807, 1397.13172221305, 8104.93589...
313861411	positive	[8163.71831638966, 737.363292309386, 3757.0401...
313861539	positive	[7805.75587107124, 974.54485808898, 3187.22594...
313861608	positive	[9349.44051500564, 1314.78991021449, 3245.6530...
313861677	positive	[8440.72712445293, 861.63534367222, 7873.85645...

	species	name \
313860745	Mus musculus	Rorb-IRES2-Cre-D;Ai14-168047.01.02.01
313861411	Mus musculus	Pvalb-IRES-Cre;Ai14-165172.05.02.01
313861539	Mus musculus	Pvalb-IRES-Cre;Ai14-165171.06.02.01
313861608	Mus musculus	Pvalb-IRES-Cre;Ai14-165874.04.02.01
313861677	Mus musculus	Rorb-IRES2-Cre-D;Ai14-166229.04.01.01

	structure_layer_name	structure_area_id	structure_area_abbrev \
313860745	5	385	VISp
313861411	4	533	VISpm
313861539	4	385	VISp
313861608	5	385	VISp
313861677	4	385	VISp

	transgenic_line	dendrite_type	apical	... overall_width \
313860745	Rorb-IRES2-Cre	spiny	truncated	... NaN
313861411	Pvalb-IRES-Cre	aspiny	NA	... NaN
313861539	Pvalb-IRES-Cre	aspiny	NA	... NaN
313861608	Pvalb-IRES-Cre	aspiny	NA	... 358.994232
313861677	Rorb-IRES2-Cre	spiny	truncated	... NaN

	scale_factor_x	scale_factor_y	scale_factor_z	soma_surface \
313860745	NaN	NaN	NaN	NaN
313861411	NaN	NaN	NaN	NaN
313861539	NaN	NaN	NaN	NaN
313861608	0.1144	0.1144	0.28	514.500853
313861677	NaN	NaN	NaN	NaN

	superseded	tags	total_length \
313860745	NaN	NaN	NaN
313861411	NaN	NaN	NaN
313861539	NaN	NaN	NaN
313861608	False	3D Neuron Reconstruction morphology	3538.87678
313861677	NaN	NaN	NaN

	total_surface	total_volume
313860745	NaN	NaN
313861411	NaN	NaN

313861539	NaN	NaN
313861608	6041.095769	945.09308
313861677	NaN	NaN

[5 rows x 44 columns]

3.6 Data Analysis & Results: Morphology

3.6.1 Spiny Neurons

We compared the differences in morphology of the spiny neurons in the 2/3 layer and the 5th layer of the primary visual cortex of mice. We did this by plotting 5 boxplots comparing the max Euclidean distance, number of stems, number of bifurcations, average contractions, and parent:daughter ratios between the 2/3 layer and the 5th layer.

```
[52]: # Set up a figure with 5 subplots
fig,ax = plt.subplots(1,5,figsize=(20,5))

VISp_df = mouse_morph_df[mouse_morph_df['structure_area_abbrev']=='VISp']
spiny_df = VISp_df[VISp_df['dendrite_type']=='spiny']

mouse_2_3_df = spiny_df[spiny_df['structure_layer_name']=='2/3']
mouse_5_df = spiny_df[spiny_df['structure_layer_name']=='5']

#Max Euclidean Distance, Number of Stems, Number of Bifurcations, Average
↳Contraction, Parent:Daughter

#On the first graph, create the box plot of Max Euclidean Distance
max_euclidean_distance_mouse_2_3_ft = mouse_2_3_df['max_euclidean_distance'].
↳dropna()
max_euclidean_distance_mouse_5_ft = mouse_5_df['max_euclidean_distance'].
↳dropna()

max_euclidean_distance_data =
↳[max_euclidean_distance_mouse_2_3_ft,max_euclidean_distance_mouse_5_ft]

ax[0].boxplot(max_euclidean_distance_data)
ax[0].set_xticklabels(['Layer 2/3','Layer 5'])
ax[0].set_ylabel('Max Euclidean Distance (m)')
ax[0].set_title('Boxplot of Max Euclidean Distance\n in Spiny Neurons')

#On the second graph, create the box plot of Number of Stems
number_stems_mouse_2_3_ft = mouse_2_3_df['number_stems'].dropna()
number_stems_mouse_5_ft = mouse_5_df['number_stems'].dropna()

number_stems_data = [number_stems_mouse_2_3_ft,number_stems_mouse_5_ft]
```

```

ax[1].boxplot(number_stems_data)
ax[1].set_xticklabels(['Layer 2/3', 'Layer 5'])
ax[1].set_ylabel('Number of Stems')
ax[1].set_title('Boxplot of Number of Stems\n in Spiny Neurons')

#On the third graph, create the box plot of the number of bifurcations
number_bifurcations_mouse_2_3_ft = mouse_2_3_df['number_bifurcations'].dropna()
number_bifurcations_mouse_5_ft = mouse_5_df['number_bifurcations'].dropna()

number_bifurcations_data = □
→ [number_bifurcations_mouse_2_3_ft, number_bifurcations_mouse_5_ft]

ax[2].boxplot(number_bifurcations_data)
ax[2].set_xticklabels(['Layer 2/3', 'Layer 5'])
ax[2].set_ylabel('Number of Bifurcations')
ax[2].set_title('Boxplot of Number of Bifurcations\n in Spiny Neurons')

#On the fourth graph, create the box plot of average contractions
average_contraction_mouse_2_3_ft = mouse_2_3_df['average_contraction'].dropna()
average_contraction_mouse_5_ft = mouse_5_df['average_contraction'].dropna()

average_contraction_data = □
→ [average_contraction_mouse_2_3_ft, average_contraction_mouse_5_ft]

ax[3].boxplot(average_contraction_data)
ax[3].set_xticklabels(['Layer 2/3', 'Layer 5'])
ax[3].set_ylabel('Average Contraction')
ax[3].set_title('Boxplot of the Average Contraction\n in Spiny Neurons')

#On the fifth graph, create the box plot of Parent-daughter ratio
average_parent_daughter_ratio_mouse_2_3_ft = □
→ mouse_2_3_df['average_parent_daughter_ratio'].dropna()
average_parent_daughter_ratio_mouse_5_ft = □
→ mouse_5_df['average_parent_daughter_ratio'].dropna()

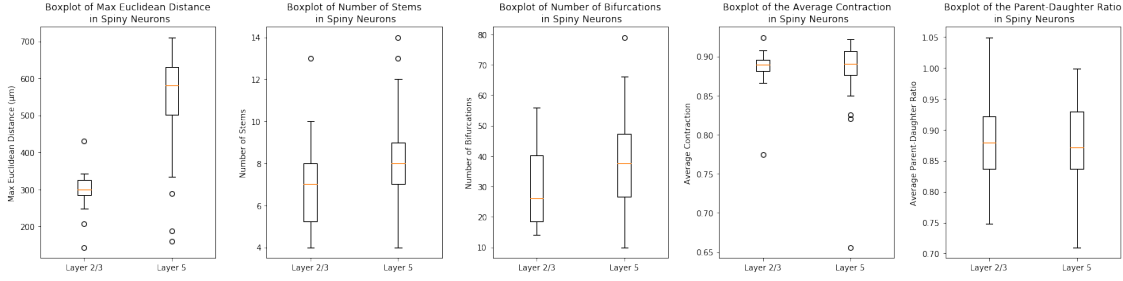
average_parent_daughter_ratio_data = □
→ [average_parent_daughter_ratio_mouse_2_3_ft, average_parent_daughter_ratio_mouse_5_ft]

ax[4].boxplot(average_parent_daughter_ratio_data)
ax[4].set_xticklabels(['Layer 2/3', 'Layer 5'])
ax[4].set_ylabel('Average Parent-Daughter Ratio')
ax[4].set_title('Boxplot of the Parent-Daughter Ratio\n in Spiny Neurons')

plt.tight_layout()

plt.show()

```



In this next step, we wanted to look at the statistical differences between layer 2/3 and layer 5. This was accomplished by using a skew test to determine whether or not the distribution for each of the morphological features had a skewed or a normal distribution. If there was a skewed distribution the Mann Whitney test was used. If there was a normal distribution, a t-test was used. The Bonferroni correction was used to correct for multiple comparisons. Thus p-values above 0.01 are not considered significant.

```
[53]: #Statistical Analysis of Max Euclidean Distance
max_euclidean_distance_mouse_2_3_skewed_stats,
↳max_euclidean_distance_mouse_2_3_skewed = stats.
↳skewtest(max_euclidean_distance_mouse_2_3_ft)
max_euclidean_distance_mouse_5_skewed_stats,
↳max_euclidean_distance_mouse_5_skewed = stats.
↳skewtest(max_euclidean_distance_mouse_5_ft)
print('The max Euclidean distance in Spiny Neurons has a ')
if max_euclidean_distance_mouse_2_3_skewed > 0.05 and
↳max_euclidean_distance_mouse_5_skewed > 0.05:
    print('normal Distributions. The T-test reports:')
    print(stats.
↳ttest_ind(max_euclidean_distance_mouse_2_3_ft,max_euclidean_distance_mouse_5_ft))
    print()
else:
    print('skewed Distribution. The Mann Whitney test results are:')
    print(stats.
↳mannwhitneyu(max_euclidean_distance_mouse_2_3_ft,max_euclidean_distance_mouse_5_ft))
    print()

#Statistical Analysis of the Number of Stems
number_stems_mouse_2_3_skewed_stats, number_stems_mouse_2_3_skewed = stats.
↳skewtest(number_stems_mouse_2_3_ft)
number_stems_mouse_5_skewed_stats, number_stems_mouse_5_skewed = stats.
↳skewtest(number_stems_mouse_5_ft)
print('The number of stems in Spiny Neurons has a')
if number_stems_mouse_2_3_skewed > 0.05 and number_stems_mouse_5_skewed > 0.05:
    print('normal distributions. The T-test reports:')
    print(stats.ttest_ind(number_stems_mouse_2_3_ft,number_stems_mouse_5_ft))
```

```

    print()
else:
    print('skewed distribution. The Mann Whitney test results are:')
    print(stats.mannwhitneyu(number_stems_mouse_2_3_ft,number_stems_mouse_5_ft))
    print()

#Statistical Analysis of the Number of Bifurcations
number_bifurcations_mouse_2_3_skewed_stats,
    ↳number_bifurcations_mouse_2_3_skewed = stats.
    ↳skewtest(number_bifurcations_mouse_2_3_ft)
number_bifurcations_mouse_5_skewed_stats, number_bifurcations_mouse_5_skewed =
    ↳stats.skewtest(number_bifurcations_mouse_5_ft)
print('The number of bifurcations in Spiny Neurons has a')
if number_bifurcations_mouse_2_3_skewed > 0.05 and
    ↳number_bifurcations_mouse_5_skewed > 0.05:
    print('normal distributions. The T-test reports:')
    print(stats.
    ↳ttest_ind(number_bifurcations_mouse_2_3_ft,number_bifurcations_mouse_5_ft))
    print()
else:
    print('skewed distribution. The Mann Whitney test results are:')
    print(stats.
    ↳mannwhitneyu(number_bifurcations_mouse_2_3_ft,number_bifurcations_mouse_5_ft))
    print()

#Statistical Analysis of Average Contractions
average_contraction_mouse_2_3_skewed_stats,
    ↳average_contraction_mouse_2_3_skewed = stats.
    ↳skewtest(average_contraction_mouse_2_3_ft)
average_contraction_mouse_5_skewed_stats, average_contraction_mouse_5_skewed =
    ↳stats.skewtest(average_contraction_mouse_5_ft)
print('The average contractions in Spiny Neurons has a')
if average_contraction_mouse_2_3_skewed > 0.05 and
    ↳average_contraction_mouse_5_skewed > 0.05:
    print('normal distributions. The T-test reports:')
    print(stats.
    ↳ttest_ind(average_contraction_mouse_2_3_ft,average_contraction_mouse_5_ft))
    print()
else:
    print('skewed distribution. The Mann Whitney test results are:')
    print(stats.
    ↳mannwhitneyu(average_contraction_mouse_2_3_ft,average_contraction_mouse_5_ft))
    print()

#Statistical Analysis of Parent Daughter ratio

```

```

average_parent_daughter_ratio_mouse_2_3_skewed_stats,
↳average_parent_daughter_ratio_mouse_2_3_skewed = stats.
↳skewtest(average_parent_daughter_ratio_mouse_2_3_ft)
average_parent_daughter_ratio_mouse_5_skewed_stats,
↳average_parent_daughter_ratio_mouse_5_skewed = stats.
↳skewtest(average_parent_daughter_ratio_mouse_5_ft)
print('The parent-daughter ratio in Spiny Neurons has a')
if average_parent_daughter_ratio_mouse_2_3_skewed > 0.05 and
↳average_parent_daughter_ratio_mouse_5_skewed > 0.05:
    print('normal distributions. The T-test reports:')
    print(stats.
↳ttest_ind(average_parent_daughter_ratio_mouse_2_3_ft,average_parent_daughter_ratio_mouse_5_
    print()
else:
    print('skewed distribution. The Mann Whitney test results are:')
    print(stats.
↳mannwhitneyu(average_parent_daughter_ratio_mouse_2_3_ft,average_parent_daughter_ratio_mouse
    print()

```

The max Euclidean distance in Spiny Neurons has a skewed Distribution. The Mann Whitney test results are:
MannwhitneyuResult(statistic=62.0, pvalue=1.081013773999717e-10)

The number of stems in Spiny Neurons has a normal distributions. The T-test reports:
Ttest_indResult(statistic=-2.4793799172611273, pvalue=0.015159884635197166)

The number of bifurcations in Spiny Neurons has a normal distributions. The T-test reports:
Ttest_indResult(statistic=-2.2573413349465397, pvalue=0.026581324000643478)

The average contractions in Spiny Neurons has a skewed distribution. The Mann Whitney test results are:
MannwhitneyuResult(statistic=676.0, pvalue=0.3927398334442342)

The parent-daughter ratio in Spiny Neurons has a normal distributions. The T-test reports:
Ttest_indResult(statistic=0.2558573674945982, pvalue=0.7986864840862002)

We can see that of the 5 features examined, number of stems, number of bifurcations, and the parent-daughter ratio were the ones with normal distributions. Max Euclidean distance and average contractions had skewed distributions. There was a significant difference (p-value < 0.01) between the two layers only for max Euclidean distance.

3.6.2 Aspiny Neurons

We performed the same analysis of morphology in aspiny neurons, plotting and performing statistical analyses on the max Euclidean distance, number of stems, number of bifurcations, average contractions, and parent:daughter ratios between the aspiny cells in layers 2/3 and layer 5.

```
[54]: # Isolate layer 2/3 and layer 5 into separate dataframes
aspiny_df = VISp_df[VISp_df['dendrite_type']=='aspiny']
aspiny_23_df = aspiny_df[aspiny_df['structure_layer_name']=='2/3']
aspiny_5_df = aspiny_df[aspiny_df['structure_layer_name']=='5']

# List of morphological features examined
features = ['max_euclidean_distance', 'number_stems', 'number_bifurcations', 'average_contraction', 'average_parent_daughter_ratio']

# Create a function that will plot each feature and perform statistical tests
def test(feature, i):
    global ax

    # Prepare data for boxplot
    layer_23 = aspiny_23_df[feature].dropna()
    layer_5 = aspiny_5_df[feature].dropna()
    data = [layer_23, layer_5]

    # Clean up feature name for plots
    if 'number' in feature:
        temp = feature.replace('number_', 'number of ')
    else:
        temp = feature
    temp2 = temp.replace('_', ' ').title()

    # Plot feature
    ax[i].boxplot(data)
    ax[i].set_xticklabels(['Layer 2/3', 'Layer 5'])
    ax[i].set_ylabel(temp2)
    ax[i].set_title('Boxplot of ' + temp2 + '\nin Aspiny Neurons')

    # Statistical Analysis for feature
    layer_23_skewed_stats, layer_23_skewed = stats.skewtest(layer_23)
    layer_5_skewed_stats, layer_5_skewed = stats.skewtest(layer_5)
    print('For ' + temp2 + ':')
    if layer_23_skewed > 0.05 and layer_5_skewed > 0.05:
        print('Normal Distributions. The T-test reports:')
        print(stats.ttest_ind(layer_23, layer_5))
        print()
    else:
        print('Skewed Distribution. The Mann Whitney test results are:')
```

```

print(stats.mannwhitneyu(layer_23, layer_5))
print()

# Plot and show statistical analyses for all features
fig,ax = plt.subplots(1,5,figsize=(20,5))
i = 0
for feature in features:
    test(feature, i)
    i += 1
plt.tight_layout()
plt.show()

```

For Max Euclidean Distance:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=1575.0, pvalue=0.008739446694131887)

For Number Of Stems:

Normal Distributions. The T-test reports:

Ttest_indResult(statistic=-2.682372949820558, pvalue=0.008282643455912488)

For Number Of Bifurcations:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=1108.0, pvalue=2.3006690355975062e-06)

For Average Contraction:

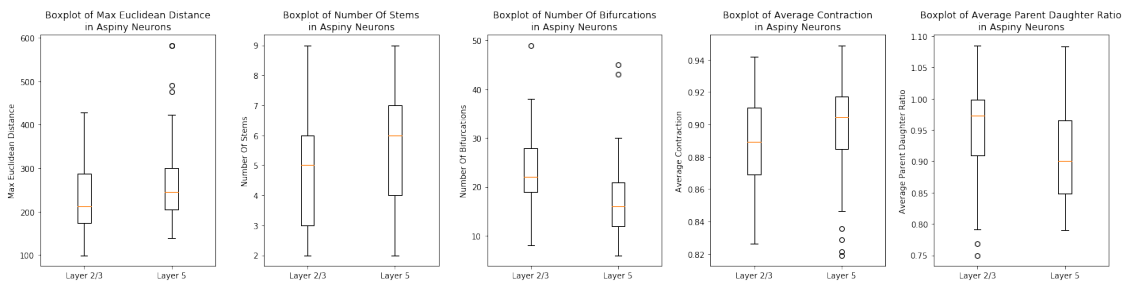
Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=1506.0, pvalue=0.003451326764634692)

For Average Parent Daughter Ratio:

Skewed Distribution. The Mann Whitney test results are:

MannwhitneyuResult(statistic=1473.5, pvalue=0.0021284374039641464)



For aspy neurons, only the number of stems followed a normal distribution; all other features showed skewed distributions. Every feature examined showed a significant difference ($p < 0.01$) between layers 2/3 and layer 5.

3.7 Dimensionality Reduction: Principal Component Analysis

3.7.1 Morphology PCA (Spiny)

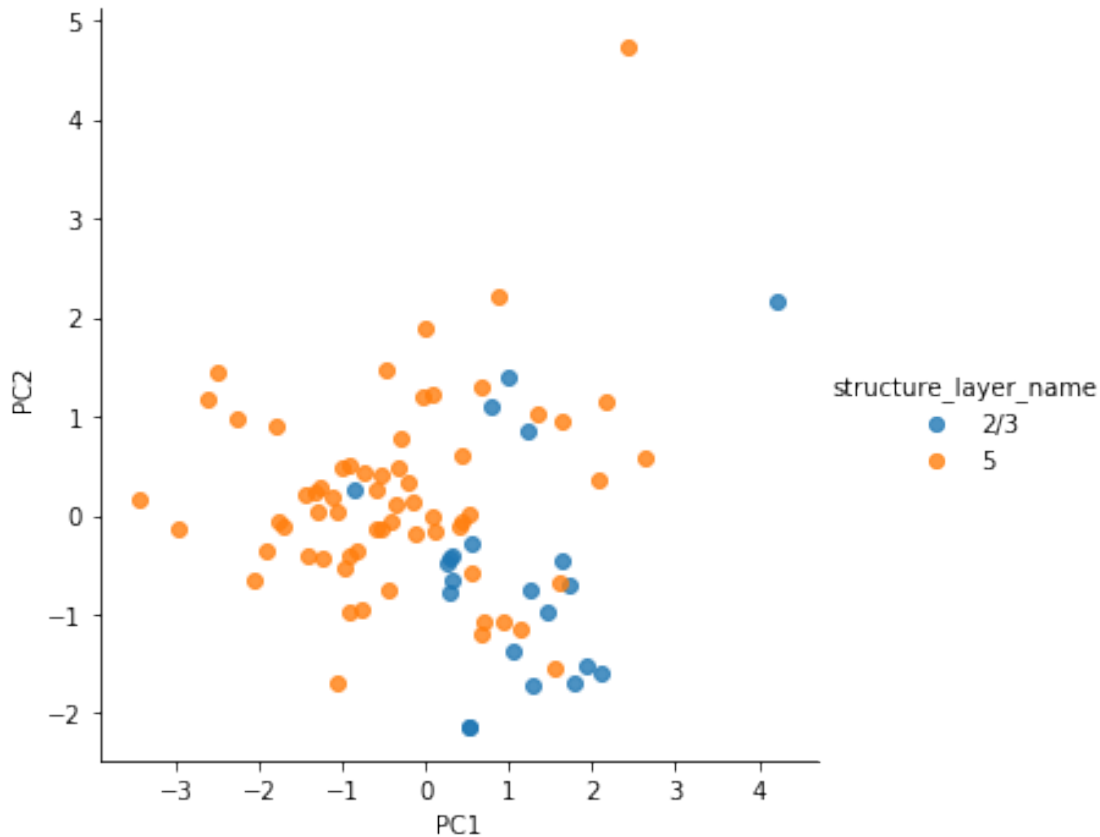
```
[55]: #Preparing the dataframe for PCA
pc_mouse_df_2 = pd.concat([mouse_2_3_df, mouse_5_df], ignore_index=True)

#Preparing the dataset for analysis with the 5 selected components
x_mouse_2 =
    ↪pc_mouse_df_2[['average_parent_daughter_ratio', 'average_contraction', 'number_bifurcations',
x_mouse_2 = x_mouse_2.dropna(axis=0).dropna(axis=1)

#Normalizing the data
x_mouse_2 = (x_mouse_2 - x_mouse_2.mean())/x_mouse_2.std()
#Preparing another dataframe needed for marking plot by layer
pc_simple_2 = x_mouse_2.copy()
pc_simple_2['structure_layer_name'] = pc_mouse_df_2['structure_layer_name']

[56]: #Initiating a PCA with 5 components
pca = PCA()
x_mouse_2_pca = pca.fit_transform(x_mouse_2)

#Plotting the PCA
pc_simple_2['PC1'] = x_mouse_2_pca[:, 0]
pc_simple_2['PC2'] = x_mouse_2_pca[:, 1]
pc_simple_2['PC3'] = x_mouse_2_pca[:, 2]
sns.lmplot("PC1", "PC2", hue='structure_layer_name', data=pc_simple_2,
    ↪fit_reg=False)
plt.show()
```

On the plot, we see the data marked by layer. The data points do not cluster in any noticeable way and doesn't seem to be correlated strongly with the layer. We then again analyse the features that contribute to individual PCs.

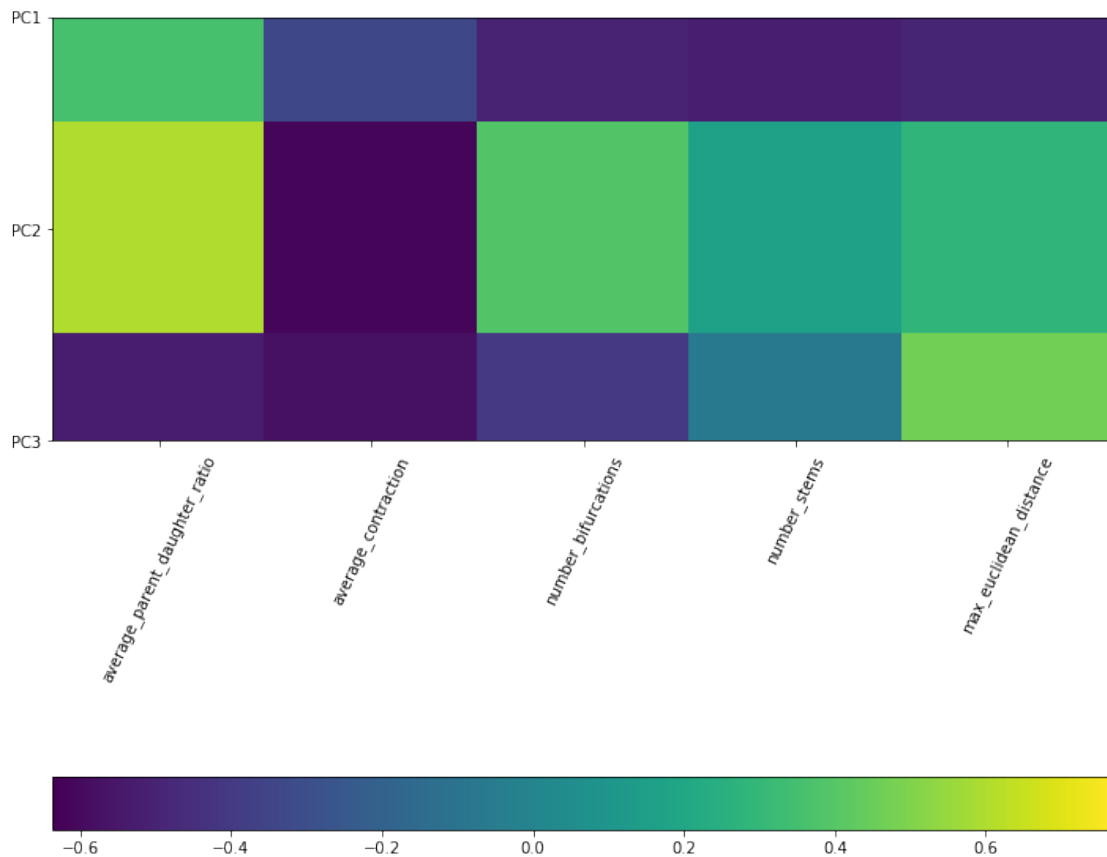
```
[57]: #Calculating the explained variance for each PC
ex_variance = np.var(x_mouse_2_pca,axis=0)
ex_variance_ratio = ex_variance/np.sum(ex_variance)
print(ex_variance_ratio)
```

```
[0.38199757 0.23148331 0.15145759 0.1325818  0.10247973]
```

The first three PCs explain over 75% of the variance.

```
[58]: #Plotting a figure explaining which features contribute to each PC
plt.figure(figsize=(10, 20))
plt.imshow(pca.components_, cmap='viridis',)
plt.yticks([0,1,2], ['PC1', 'PC2', 'PC3'], fontsize=10)
plt.colorbar(orientation='horizontal')
plt.tight_layout()
plt.xticks(range(len(x_mouse_2.columns)), x_mouse_2.columns, rotation=65)
```

```
plt.show()
```



Both PC1 and PC2 are mostly made up of the average parent-daughter ratio. It is worth to notice that it is not something we found to be significantly different between the layers.

3.7.2 Morphology PCA (Aspiny)

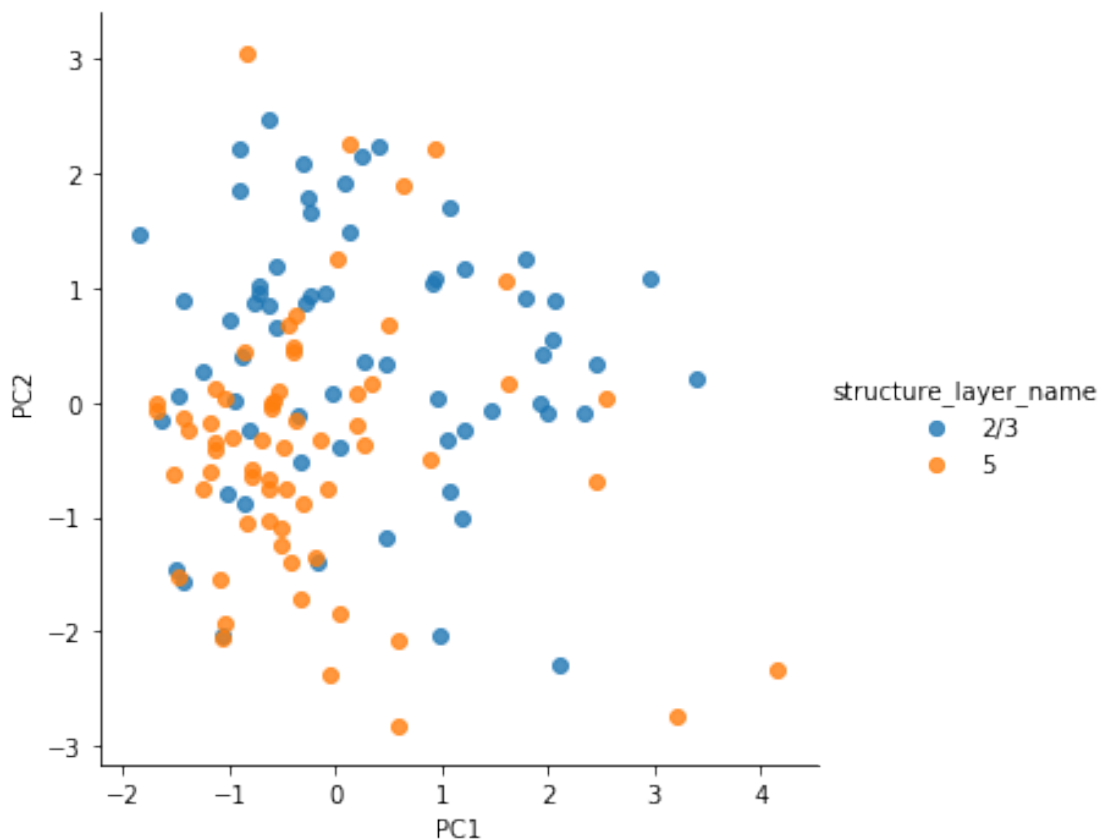
```
[59]: #Preparing the dataframe for PCA
pc_mouse_df_3 = pd.concat([aspiny_23_df, aspiny_5_df])

#Preparing the dataset for analysis with the 5 selected components
x_mouse_3 =
↳pc_mouse_df_3[['average_parent_daughter_ratio', 'average_contraction', 'number_bifurcations',
x_mouse_3 = x_mouse_3.dropna(axis=0).dropna(axis=1)

#Normalizing the data
x_mouse_3 = (x_mouse_3 - x_mouse_3.mean())/x_mouse_3.std()
#Preparing another dataframe needed for marking plot by layer
pc_simple_3 = x_mouse_3.copy()
```

```
pc_simple_3['structure_layer_name'] = pc_mouse_df_3['structure_layer_name']
```

```
[60]: #Initiating a PCA with 5 components
pca = PCA()
x_mouse_3_pca = pca.fit_transform(x_mouse_3)
#Plotting the PCA
pc_simple_3['PC1'] = x_mouse_3_pca[:, 0]
pc_simple_3['PC2'] = x_mouse_3_pca[:, 1]
pc_simple_3['PC3'] = x_mouse_3_pca[:, 2]
sns.lmplot("PC1", "PC2", hue='structure_layer_name', data=pc_simple_3,
           fit_reg=False)
plt.show()
```



Again, we see no defined clusters and no layer-specific division between the datapoints.

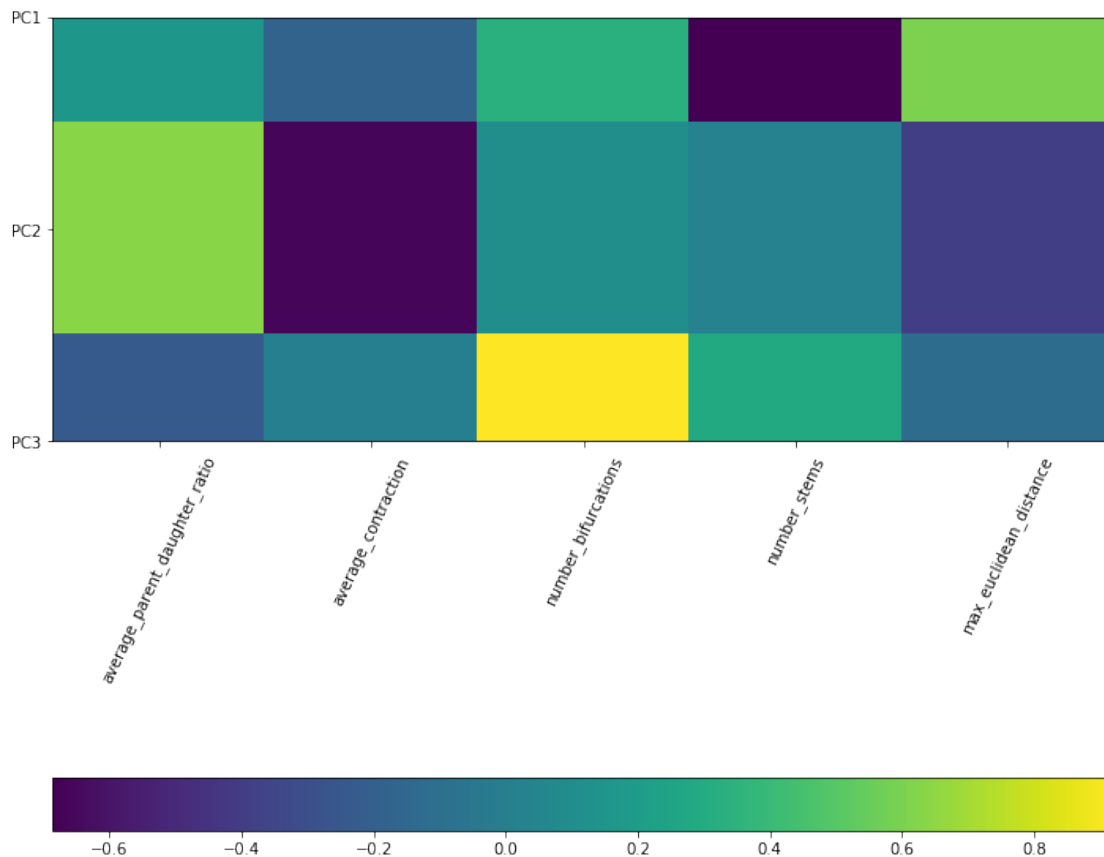
```
[61]: #Calculating the explained variance for each PC
ex_variance = np.var(x_mouse_3_pca,axis=0)
ex_variance_ratio = ex_variance/np.sum(ex_variance)
print(ex_variance_ratio)
```

```
[0.29635111 0.28375443 0.18866539 0.13170496 0.0995241 ]
```

Over 70% of the variance can be explained by the first three PCs.

```
[62]: #Plotting a figure explaining which features contribute to each PC
plt.figure(figsize=(10, 20))
plt.imshow(pca.components_,cmap='viridis',)
plt.yticks([0,1,2],['PC1','PC2','PC3'],fontsize=10)
plt.colorbar(orientation='horizontal')
plt.tight_layout()
plt.xticks(range(len(x_mouse_3.columns)),x_mouse_3.columns,rotation=65)

plt.show()
```



PC1 seems to be mostly contributed to by the max Euclidean distance, PC2 by the parent daughter ratio while PC3 is strongly defined by the number of bifurcations.

3.8 Setup: ISH

The Allen Brain Atlas API is used to pull differential gene expression from the mouse in situ hybridization (ISH) data. This compares a set of target structures to a set of contrast structures and gives a ‘fold change’ value that reveals the expression of reported genes in the target structures

relative to the contrast structures. Here, the target structures are layers 2/3 of the mouse visual cortex, and the contrast structures the layer 5 visual cortex areas. The data pulled from the API is then converted into a Pandas Dataframe so that it can more easily be viewed and analyzed.

```
[63]: import requests
import json
```

3.9 Data Wrangling: ISH

```
[64]: # Set up API to retrieve mouse ISH data
service = 'http://api.brain-map.org/api/v2/data/query.json?criteria='
structures_1 = ['Primary visual area, layer 2/3']
structures_2 = ['Primary visual area, layer 5']
ontology = 'Mouse Brain Atlas'

# Get IDs for examined structures
ids_1 = []
ids_2 = []
for item in structures_1:
    structure_id = requests.get('%smodel::Structure,\
                                rma::criteria,[name$il\\'%s\\']',ontology[name$eq\\'%s\\']' %_\
                                ↪(service,item,ontology)).json()['msg'][0]['id']
    ids_1.append(str(structure_id))
for item in structures_2:
    structure_id = requests.get('%smodel::Structure,\
                                rma::criteria,[name$il\\'%s\\']',ontology[name$eq\\'%s\\']' %_\
                                ↪(service,item,ontology)).json()['msg'][0]['id']
    ids_2.append(str(structure_id))

# Create a single string for each set of structure IDs to use in API
temp1 = ''
temp2 = ''
for i in ids_1:
    temp1 += i + ','
for i in ids_2:
    temp2 += i + ','
structures1 = temp1[0:-1]
structures2 = temp2[0:-1]

# Get results for differential search and put into a pandas dataframe
all_results = []
test = True
start_row = 0
# API is only able to retrieve 2000 rows at once, loop until none remain
while test:
```

```

    result = requests.get('%sservice::
↪mouse_differential[set$eq\'mouse\'] [structures1$eq\'%s\'] [structures2$eq\'%s\'] [start_row$
↪\
                                % (service, structures1, structures2,
↪str(start_row))).json()
    if result['success'] == True:
        all_results += result['msg']
        start_row += 2000
    else:
        test = False
mouse_differential = pd.DataFrame(all_results)
mouse_differential

```

```

[64]:
      id      name  gene-id gene-symbol \
0    100055069  RP_080110_01_A05    11714    Atp1a1
1     73423932  RP_051019_03_B09    18312     Pcp4
2     545100    RP_040825_02_D01    46815    Pgm2l1
3    112202624  RP_110421_01_F02    60343     Klf7
4       772    RP_Baylor_102947    18312     Pcp4
...
9503   70473879  RP_050614_04_D10    76555     Egl1n1
9504   74562600  RP_060208_02_G09    93352    Slc9a5
9505   71161042  RP_050810_02_E08    85210     Dgkb
9506     2325    RP_Baylor_102862    16144     Irx1
9507   71021147  RP_050526_02_G10    74949     Cebpe

```

```

                                gene-name  entrez-id  chromosome \
0      ATPase, Na+/K+ transporting, alpha 1 polypeptide    11928         3
1                Purkinje cell protein 4    18546        16
2            phosphoglucomutase 2-like 1    70974         7
3      Kruppel-like factor 7 (ubiquitous)    93691         1
4                Purkinje cell protein 4    18546        16
...
9503                EGL nine homolog 1 (C. elegans)    112405         8
9504  solute carrier family 9 (sodium/hydrogen excha...    277973         8
9505                diacylglycerol kinase, beta    217480        12
9506      Iroquois related homeobox 1 (Drosophila)    16371        13
9507  CCAAT/enhancer binding protein (C/EBP), epsilon    110794        14

```

```

plane-of-section specimen-id fold-change target-sum contrast-sum \
0      sagittal      n/a      10.208    1586.732    171.975
1      sagittal      n/a      8.739    1801.084    206.098
2      sagittal      n/a      7.103    1802.017    284.020
3      coronal      n/a      6.891    920.788    151.672
4      coronal      n/a      6.556    1130.932    194.852
...
9503   sagittal      n/a      0.433    182.888    504.841

```

9504	sagittal	n/a	0.360	120.634	390.811
9505	sagittal	n/a	0.354	158.971	514.402
9506	coronal	n/a	0.343	138.575	455.855
9507	sagittal	n/a	0.307	85.246	347.034

	num-target-samples	num-contrast-samples
0	94	104
1	106	106
2	92	103
3	74	84
4	108	122
...
9503	72	86
9504	90	105
9505	90	103
9506	108	122
9507	76	95

[9508 rows x 14 columns]

3.10 Data Analysis & Results: ISH

With the data now in the proper format, the differential gene expression can be visualized as a histogram to observe its distribution. If gene expression is the same between the different layers, we could expect the distribution to follow a normal gaussian distribution centered at 1 with a width of 1 since the ratio of expression cannot be less than 0. The differential gene expression will be plotted against a random normal distribution to determine whether there are notable differences in gene expression across the layers.

```
[65]: # Fold-change values are strings, get values as floats
fold_change = []
ind = 0
for i in mouse_differential['fold-change']:
    fold_change.append(float(i))
    ind += 1

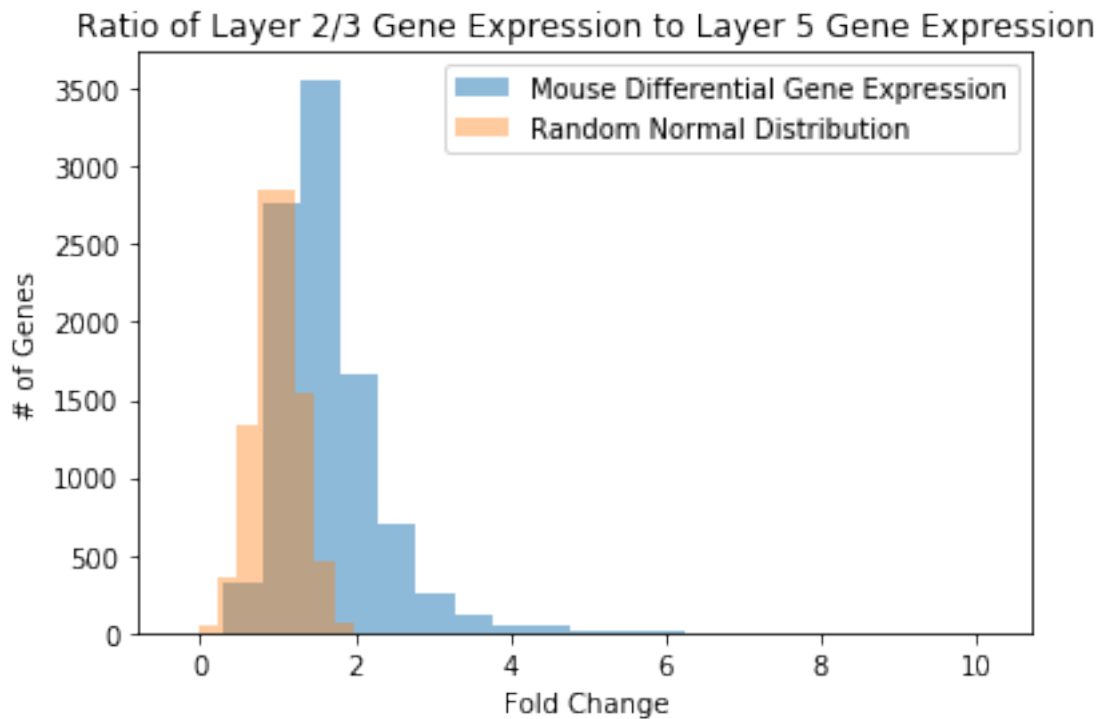
# Create a random normal distribution with the same sample size as the
↳ differential data
sample_mean, sample_sigma = 1, .3
sample_norm = np.random.normal(sample_mean, sample_sigma, len(fold_change))

# Plot the distributions
fig, ax = plt.subplots()
ax.hist(fold_change, bins=20, alpha=0.5, label='Mouse Differential Gene
↳ Expression')
ax.hist(sample_norm, alpha=0.4, label='Random Normal Distribution')
```

```

ax.legend()
ax.set_xlabel('Fold Change')
ax.set_ylabel('# of Genes')
ax.set_title('Ratio of Layer 2/3 Gene Expression to Layer 5 Gene Expression')
plt.show()

```



```

[66]: print(stats.skewtest(fold_change))
      print(stats.mannwhitneyu(fold_change,sample_norm))

```

```

SkewtestResult(statistic=55.1908308926386, pvalue=0.0)
MannwhitneyuResult(statistic=14101031.0, pvalue=0.0)

```

The plot clearly shows that the distribution of differential gene expression is very different from a normal distribution, and a skewness test followed by hypothesis test confirms that the two distributions are not equal. If we define a significant difference of expression as 50%, i.e. less than 0.5 or greater than 1.5 fold change, we can look at the percentage of genes that have significantly different levels of expression.

```

[67]: similar = []
      greater = []
      less = []

      for item in fold_change:
          if item < 0.50:

```



```

        less.append(item)
    elif item < 1.5:
        similar.append(item)
    else:
        greater.append(item)

percent_less = len(less)/len(fold_change)
percent_similar = len(similar)/len(fold_change)
percent_greater = len(greater)/len(fold_change)

print('% less:    ', percent_less*100, '\n% similar:', percent_similar*100, '\n%
→greater:', percent_greater*100)

```

```

% less:    0.18931426167437948
% similar: 49.90534286916281
% greater: 49.90534286916281

```

Nearly 50% of genes tested are expressed at much different levels in layer 2/3 compared to layer 5, which suggests that there are indeed transcriptomic differences between different layers of the mouse primary visual cortex.

4 Conclusion & Discussion

The electrophysiology analysis of the two layers revealed that, out of the 9 properties analysed, the only significant differences were between the rheobase, adaptation index, membrane time constant, ramp spike time and the resting potentials. The resting potential for layer 2/3 was found to be lower than layer 5, which is consistent with other findings (Senzai et al., 2019) and implies why layer 2/3 neurons have been previously found to have the lowest firing rates while layer 5 neurons have the highest firing rates. The rheobase, lower for layer 5, further confirms that, suggesting higher excitability of layer 5 neurons. Our findings that the average ISI is higher for layer 5 is also consistent with Senzai's group's findings.

The morphology analysis for the spiny neurons only found a significant difference in max Euclidean distance while analysis for aspiny neurons found significant differences between the layers for all features analysed: max Euclidean distance, number of stems, number of bifurcations, average contractions, and parent:daughter ratios. Classically, spiny neurons are considered to be excitatory while aspiny neurons are generally classified as inhibitory. However, previous research has shown that dendritic spines are also found in some inhibitory neurons and suggests that spines are a universal structure that serve as flexible elements in synaptic connections (Scheuss & Bonhoeffer, 2013). This universality suggests that spiny neurons are a relatively heterogenous group, as reflected by our results where nearly all features of spiny neurons examined were not considered to be different across layers. Layer 5 did however show a significantly increased max Euclidean distance, defined as the maximum straight line distance from the soma of the neuron to the furthest node, compared to layer 2/3. The increased length of the spiny neurons in this layer may potentially play a role in any potential functional differences between the two layers. Meanwhile, aspiny neurons showed differences across all features examined between layers 2/3 and 5. Unfortunately, there seems to be a lack of research on specific morphological features of neurons and their implications on neuronal

function. However, some studies have shown that mouse visual cortex contains circuits based upon specific within-layer connectivity to aspiny neurons (Bopp et al., 2014). The differences in the morphology of aspiny neurons between the layers could possibly reflect these layer-specific circuits, which in turn might explain some of the functional differences previously explained between layer 2/3 and layer 5.

We hypothesised that the electrophysiological and morphological differences could be explained by the differences in gene expression between those two layers - hence, we plotted the fold change of gene expression as a histogram comparing the distribution to a normal distribution, assuming a normal distribution would imply the fold change is insignificant. The distributions were clearly not equal, suggesting that there is a difference in gene expression. We then looked at the percentage of genes that show at least 50% change in gene expression and found nearly 50% of genes differ, implying this difference might indeed underlie the electrophysiological and morphological differences.

The genes which differ the most between the layers include *Atp1a1* which encodes a Na⁺/K⁺ ATPase. This could explain some of the differences in electrophysiology as the pump is responsible for maintaining the electrochemical sodium/potassium gradient which is essential in determining the electrical excitability of neurons. It might possibly be the cause of the differences between the rheobase values as it is a measure of membrane potential excitability and the average resting membrane potential differences. Another gene is *Pcp4* which is a known marker for excitatory neurons of layer 5. It is a modulator of calcium-binding by calmodulin and it has been hypothesised to play a role in neuronal differentiation through activation of downstream kinase signalling pathways (Watakabe et al., 2012) which further implies the neuron type composition is different between the layers. *Crym* and *Whrn* are genes involved with vision specifically, suggesting that there are functional differences between the layers when it comes to visual processing. Overall, we see that the genes with highest fold change between the layers could explain the electrophysiological and functional differences and should be investigated further.

Our findings have a very limited scope as we not only compared 2 layers but also only 1 brain area so they might have limited significance in the greater scheme. This is due to the limited data available online in the Allen Cell Type Atlas as well as the limited time and resources that we have which don't allow us to expand this project. It would also be important in the future to analyse function of more genes and try to correlate their functions to the layers'/brain areas' functions, predominant neuronal types present as well as consider projections, target structures and so on. Again, this would require a significantly bigger project and much more data.

There is a huge diversity within the electrophysiology data with many outliers which may suggest another limitation of our approach. Clearly, each layer consists of many distinct neuronal types which implies that it could be more useful to categorise cortical neurons into types and study their roles as a part of the circuit rather than a layer, as done by Tasic et al., 2018. That approach could also possibly explain the two clusters we see in the electrophysiology PCA as they are clearly not correlated with the layer. The morphology PCA's results are also not very meaningful - maybe because the morphology dataset only had 86 spiny and 129 aspiny cells that matched out criteria. Such a small sample (compared to the actual number of neurons within each layer) doesn't give reliable results. The same could be said about the morphology analysis in general - which makes the fact that there are little outliers less significant.

Another limitation is transferability of the results to humans. However, it has been suggested that transcriptomic findings in mouse are highly relevant to human biology because of the strong similarities between the brain transcriptomes of these species' (Belgard et al., 2011) so the findings

might be informative of human visual cortex as well.

However, overall our findings seem to be consistent with the literature, suggesting that, to some extent, they are reliable. In conclusion, I think we can fairly say that there are differences between layer 2/3 and layer 5 of the mouse visual primary cortex and they might partly be explained by the gene expression differences.

4.1 Reflection

The most difficult part about the project was joining the data available online in a meaningful way to answer some interesting questions. It has often proven difficult to find common cell features criteria specified within the datasets which was crucial for comparison of any sort; early ideas for the project had to be abandoned due to the types of data that were available in the different datasets. It was also hard sometimes to understand the documentation for interacting with the specific datasets but at the same time very rewarding to figure out. The most rewarding element however, was being able to apply the skills we learned in class to perform our own exploratory analyses as opposed to simply following instructions given in assignments.