BIPN 162 Final Project Group 8-5

March 16, 2020

1 Contributions

NAME: Data extraction, wrangling and analysis

NAME: Introduction & Background, Conclusion & Discussion, Abstract, PCA analysis

NAME: Abstract, Data Analysis, Limitations, Conclusion & Discussion, Hypothesis, DAVID analysis

2 Abstract

Little is known about how these cues are processed beyond primary cortical areas, our findings may show how the genes and the connectivity outputs lead to functional specialization in higher order visual behaviors.The anterolateral area and posteromedial area in the visual cortex is believed to be responsible for the fast moving and slow moving object respectively. We want to see how these differences in gene expression, connectivity, resting membrane potential, and directional selectivity provide insight towards the functional specialization of the two regions. Using a DAVID analysis to uncover functional differentiation in the gene expressions between the two regions, the analysis showed that that the anterolateral area has very high gene expression in genes that code for neurotransmitter receptors (such as serotonin, oxytocin, dopamine, and several others) when compared to posteromedial area, while the posteromedial area had very high gene expression in genes that coded for the Golgi apparatus and the Golgi membrane, which is responsible for the packaging of proteins, and very high gene expression in the cAMP pathway. Varying very little in electrophysiological features, the AL and PM project to different regions that have different functions. These findings may show how the genes and the connectivity outputs lead to functional specialization in higher order visual behaviors.

3 Research Question

How do differences in gene expression, connectvity, resting membrane potential, and directional selectivity lead to functional specialization between the anterolateral area (AL) and the posteromedial area (PM) in higher visual areas of the mice?

4 Background and Prior Work

Uncovering the mechanisms behind the flow and processing of visual information is a difficult problem, yet one that is fundamental to understanding the sensory systems as a whole. A necessary step toward the detailed study of the visual hierarchy is a thorough characterization of boundaries and visual field representations.[1] Mouse brains are made up of many millions of cells called neurons that are interconnected to form neuronal circuits.[2] Neurons that express similar genes tend to look and have electrophysiological alike, whereas neurons that express different genes tend to be dissimilar. [2] We want to determine which genes and the resulting functional specialization are expressed in groups of neurons that represent the many cell types found in many parts of the brain, including the visual cortex.[1]

Specialized neural circuits process visual information in parallel hierarchical streams, leading to complex visual perception and behavior.[3] Distinct channels of visual information begin in the retina of the eye and synapse through the lateral geniculate nucleus to the primary visual cortex (V1), forming the building blocks for visual perception.[3] In this proposal, we are comparing the differences of the anterolateral area (AL) and the posteromedial (PM) area in higher visual areas of mice's visual cortex. Anderman et al. found that the anterolateral area (AL) of the visual cortex responsible for guiding behavior involving fast-moving stimuli and the posteromedial(PM) area helps guide behavior involving slow-moving objects. [3] Zariwala, Hatim A., et al. looked at the genetic differences in the visual cortex with cre-transgenic mice.[4] Huh, C. Y. et al show that putatively feedback neurons in layer 5(L5) in higher visual areas, AL and PM, display distinct visual properties; AL L5 feedback neurons prefer significantly lower spatial frequency compared to PM L5 feedback neurons. [6] These behaviors are critical in understanding higher-order cognition, a complex type of thinking which refers to the mental processes of reasoning, decision making, and creativity, etc.

The data sets that we are working with are Allen Brain Observatory, Allen Cell Type data, and Allen Mouse Brain Connectivity Atlas. The Allen Brain Observatory is data for how visual stimuli are represented by neural activity in the mouse visual cortex in both single cells and populations. A calcium imaging for different mouse cre lines with a calcium reporter was analyzed during exposure to five classical visual stimuli: drifting gratings, static gratings, natural scenes, natural movies, locally sparse noise. There are 30 experiments with the drifting gradient for both visual cortex regions that have data for directional selective, differential response to the direction of a visual stimulus. Glickfeld, L. et al. o used two-photon calcium imaging in mice compared visual responses in primary visual cortex(V1) and the same downstream target, AL and PM.[5] In their experiments, they presented the mice with upward or downward drifting grating at five spatial frequencies to determine directional selectivity.[6] The Allen Brain Observatory provides a dataset to survey information encoding in the visual cortex. The Allen Cell Type data is data from a single neuron from mice and humans from electrophysiological, morphological, and transcriptomic data. For electrophysiology data, the researchers did a whole-cell patch-clamp recording to find upstroke and downstroke for over 2,000 neurons. The transcriptomic data is collected using an RNA sequence of single cells. The gene transcripts are isolated from whole cells or nuclei, amplified, and sequenced, and then aligned to a reference genome. There is data for gene expression for 2,000 genes in both the anterolateral area (AL) and the posteromedial area (PM). Huh et al looked at the connevity of the region using retrograde tracing using Cre recombinase to optogenetically tagged cells. [6] The Allen Mouse Brain Connectivity is made up not only retrograde input mapping but has anterograde projection mapping , biotinylated dextran amines (BDA) vs. adeno-associated virus (rAAV)

Comparison, Transgenic characterization and anatomic references. With these datasets, we would like to see what leads to functional specialization in AL and PM.

4.1 References (include links):

(1) Wang, Q., & Burkhalter, A. (2007). Area map of mouse visual cortex. The Journal of Comparative Neurology, 502(3), 339–357. doi:10.1002/cne.21286 https://www.ncbi.nlm.nih.gov/pubmed/17366604

(2) de Vries, S.E.J., Lecoq, J.A., Buice, M.A. et al. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. Nat Neurosci 23, 138–151 (2020). https://doi.org/10.1038/s41593-019-0550-9

(3) Andermann, Mark L et al. "Functional specialization of mouse higher visual cortical areas." Neuron vol. 72,6 (2011): 1025-39. doi:10.1016/j.neuron.2011.11.013 https://www.ncbi.nlm.nih.gov/pubmed/22196337

(4) Zariwala, Hatim A., et al. "Visual tuning properties of genetically identified layer 2/3 neuronal types in the primary visual cortex of cre-transgenic mice." Frontiers in systems neuroscience 4 (2011): 162. https://www.ncbi.nlm.nih.gov/pubmed/21283555

(5) Glickfeld, L., Andermann, M., Bonin, V. et al. Cortico-cortical projections in mouse visual cortex are functionally target specific. Nat Neurosci 16, 219–226 (2013). https://doi.org/10.1038/nn.3300

(6) Huh, C. Y., Peach, J. P., Bennett, C., Vega, R. M., & Hestrin, S. (2018). Feature-Specific Organization of Feedback Pathways in Mouse Visual Cortex. Current Biology, 28(1). https://www.cell.com/current-biology/references/S0960-9822(17)31538-5#%20

(7) Lyamzin, Dmitry & Benucci, Andrea. (2018). The mouse posterior parietal cortex: Anatomy and functions. Neuroscience Research. https://www.sciencedirect.com/science/article/pii/S0168010218306102

(8) Andersen, R.A., 1997. Multimodal integration for the representation of space in the posterior parietal cortex. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences, 352(1360), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1692052/pdf/9368930.pdf

(9) Harris, J. A., Mihalas, S., Hirokawa, K. E., Whitesell, J. D., Knox, J., Bernard, A., … & Feng, D. (2018). The organization of intracortical connections by layer and cell class in the mouse brain. BioRxiv, 292961. https://www.biorxiv.org/content/10.1101/292961v1.full.pdf

(10) Mitchell, A.S., Czajkowski, R., Zhang, N., Jeffery, K. and Nelson, A.J., 2018. Retrosplenial cortex and its role in spatial cognition. Brain and neuroscience advances, 2, p.2398212818757098. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6095108/

5 Hypothesis

We hypothesize that the anterolateral area and the posteromedial area will be dissimilar in gene expression because the anterolateral area of the visual cortex should have a gene expression pattern

that will allow for it to guide behavior involving fast moving stimuli while the posteromedial area should have a gene expression pattern that will allow for it to guide behavior involving slow moving objects. We also expect the electrophysiological features of the two brain areas to be dissimilar. The resting membrane potential is expected to be different between the two areas because neurons that express different genes tend to have electrophysiological characteristics that are dissimilar. We also hypothesize that the projection matrices of the two regions will be dissimilar because they have different function specializations therefore they will have different downstream projection regions.

6 Code

6.1 Set-up

The packages and datasets required in this project is provided below

[1]:
```
# Importing required packages for the project
import io
import json
import requests
import pprint
import warnings
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import allensdk.brain_observatory.stimulus_info as stim_info

from scipy import stats
from collections import Counter
from sklearn.decomposition import PCA
from pandas.io.json import json_normalize
from scipy.spatial.distance import pdist, squareform
from allensdk.core.cell_types_cache import CellTypesCache
from allensdk.api.queries.cell_types_api import CellTypesApi
from allensdk.core.brain_observatory_cache import BrainObservatoryCache
from allensdk.brain_observatory.ecephys.ecephys_project_cache import␣
 ↪EcephysProjectCache
from allensdk.core.mouse_connectivity_cache import MouseConnectivityCache
from allensdk.api.queries.rma_api import RmaApi

# Set up style and palette for seaborn plot
sns.set(style="darkgrid")
sns.set_palette("bright")
```

Download required mouse data from Allen Cell Types Database

[2]:
```
ctc = CellTypesCache(manifest_file='cell_types/manifest.json')
mouse_df = pd.DataFrame(ctc.get_cells(species=[CellTypesApi.MOUSE]))
```

6.1.1 Import CSV including ISH gene expression data from the two structures

Approach: The query search is provided below: http://api.brain-map.org/api/v2/data/query.xml?criteria=service::mouse_differential[set$eq'mouse'][structures1$eq402][structures

Visual AnteriorLateral Area : Structure.id = 533

Visual PosteriorMedial Area : Structure.id = 402

Primary Visual Cortex: Structure.id = 385

Visual Areas: Structure.id = 669

Basic Cell groups: Structure.id = 8

Package all the api requests that will implement differential search of structures in Allen database.

[3]:
```
NL_AL_API = "http://api.brain-map.org/api/v2/data/query.json?criteria=service::
 ↪mouse_differential[set$eq'mouse'][structures1$eq533][structures2$eq385]"
NL_PM_API = "http://api.brain-map.org/api/v2/data/query.json?criteria=service::
 ↪mouse_differential[set$eq'mouse'][structures1$eq402][structures2$eq385]"
```

Request the genetic data from Allen API service, then converting all the json data into pandas dataframe

[4]:
```
NL_AL = requests.get(NL_AL_API)
NL_AL_json = NL_AL.json()
NL_AL_json_processed = NL_AL_json['msg']
new_columns = list(NL_AL_json_processed[0].keys())
NL_VISal_pd = pd.DataFrame(columns = new_columns)

for dictionaries in NL_AL_json_processed:
    gene_list = pd.DataFrame(list(dictionaries.items())).transpose()
    gene_list.columns = gene_list.loc[0]
    gene_list = pd.DataFrame(gene_list.drop(0))
    NL_VISal_pd = pd.concat([NL_VISal_pd, gene_list])

NL_PM = requests.get(NL_PM_API)
NL_PM_json = NL_PM.json()
NL_PM_json_processed = NL_PM_json['msg']
new_columns = list(NL_PM_json_processed[0].keys())
NL_VISpm_pd = pd.DataFrame(columns = new_columns)

for dictionaries in NL_PM_json_processed:
```

```
    gene_list = pd.DataFrame(list(dictionaries.items())).transpose()
    gene_list.columns = gene_list.loc[0]
    gene_list = pd.DataFrame(gene_list.drop(0))
    NL_VISpm_pd = pd.concat([NL_VISpm_pd, gene_list])
```

6.1.2 Accessing Brain Observatory data

Download a list of all targeted areas

[5]:
```
boc = BrainObservatoryCache(manifest_file='/datasets/allen-brain-observatory/
 ↪visual-coding-2p/manifest.json')
targeted_structures = boc.get_all_targeted_structures()
```

6.1.3 Accessing Mouse Connectivity Data

[6]:
```
mcc = MouseConnectivityCache()
structure_tree = mcc.get_structure_tree()
id_acronym = structure_tree.get_id_acronym_map()
```

6.2 Data Wrangling

6.2.1 Handling Genetic Data

Extract the gene symbols and related fold change from the pd dataframe

[7]:
```
NL_VISal_pd_fold = NL_VISal_pd[['gene-symbol', 'fold-change']]
NL_VISal_length = NL_VISal_pd_fold.shape[0]

NL_VISpm_pd_fold = NL_VISpm_pd[['gene-symbol', 'fold-change']]
NL_VISpm_length = NL_VISpm_pd_fold.shape[0]
```

Change all the fold change data into numeric data for later analysis

[8]:
```
NL_VISal_pd_fold.loc[:,('fold-change')] = pd.to_numeric(NL_VISal_pd_fold.loc[:
 ↪,('fold-change')])
NL_VISpm_pd_fold.loc[:,('fold-change')] = pd.to_numeric(NL_VISpm_pd_fold.loc[:
 ↪,('fold-change')])
```

```
/opt/conda/lib/python3.6/site-packages/pandas/core/indexing.py:576:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
```

```
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self.obj[item_labels[indexer[info_axis]]] = value
```

Merge mouse id dataframe with the elctrophysiology features

[9]:
```
mouse_df = mouse_df.set_index('id')
ephys_features = pd.DataFrame(ctc.get_ephys_features()).set_index('specimen_id')
mouse_ephys_df = mouse_df.join(ephys_features, how ='inner')
ephys_columns = list(ephys_features.columns)
```

6.2.2 Further selecting data for dimensionality reduction through PCA

Selecting Data for analysing Resting Potential data from cell types database

[10]:
```
visal_ephys_data = mouse_ephys_df.loc[mouse_ephys_df["structure_area_abbrev"]␣
 ↪== "VISal"].copy()
print(visal_ephys_data.shape[0])
visal_ephys_vrest = pd.DataFrame(visal_ephys_data.loc[:, 'vrest'].copy())
visal_ephys_vrest.columns = ['Vrest']
visal_column = np.repeat("VISal", visal_ephys_vrest.shape[0])
visal_ephys_vrest['Brain Region'] = visal_column
visal_ephys_data['Brain Region'] = visal_column

vispm_ephys_data = mouse_ephys_df.loc[mouse_ephys_df["structure_area_abbrev"]␣
 ↪== "VISpm"].copy()
print(vispm_ephys_data.shape[0])
vispm_ephys_vrest = pd.DataFrame(vispm_ephys_data.loc[:, 'vrest'].copy())
vispm_ephys_vrest.columns = ['Vrest']
vispm_column = np.repeat("VISpm", vispm_ephys_vrest.shape[0])
vispm_ephys_vrest['Brain Region'] = vispm_column
vispm_ephys_data['Brain Region'] = vispm_column

vrest = pd.concat([visal_ephys_vrest, vispm_ephys_vrest])
vis_alpm_data = pd.concat([visal_ephys_data, vispm_ephys_data])
```

```
17
89
```

Pre-processing of the ephys-data dataframe before running PCA 1. Seperating the dataframe into VISal and VISpm cells 2. Dropping columns that are non-numerical 3. Dropping columns that have N/A entries

[11]:
```
cell_list = ['reporter_status', 'cell_soma_location', 'species', 'name',
       'structure_layer_name', 'structure_area_id', 'structure_area_abbrev',
       'dendrite_type', 'apical', 'reconstruction_type',
       'disease_state', 'donor_id', 'structure_hemisphere',
       'normalized_depth']
visal_ephys_data = visal_ephys_data.drop(cell_list, axis = 1)
```

```
vispm_ephys_data = vispm_ephys_data.drop(cell_list, axis = 1)

visal_ephys_transgenic = visal_ephys_data.copy()
print(len(visal_ephys_transgenic['transgenic_line']))
vispm_ephys_transgenic = vispm_ephys_data.copy()
print(len(vispm_ephys_transgenic['transgenic_line']))

visal_ephys_data = visal_ephys_data.drop(['transgenic_line'], axis = 1)
vispm_ephys_data = vispm_ephys_data.drop(['transgenic_line'], axis = 1)

ephys_list =['adaptation', 'avg_isi', 'electrode_0_pa', 'f_i_curve_slope',
       'fast_trough_t_long_square', 'fast_trough_t_ramp',
       'fast_trough_t_short_square', 'fast_trough_v_long_square',
       'fast_trough_v_ramp', 'fast_trough_v_short_square',
       'input_resistance_mohm', 'latency',
       'peak_t_long_square', 'peak_t_ramp', 'peak_t_short_square',
       'peak_v_long_square', 'peak_v_ramp', 'peak_v_short_square',
       'rheobase_sweep_id', 'rheobase_sweep_number', 'ri', 'sag', 'seal_gohm',
       'threshold_i_long_square', 'threshold_i_ramp',
       'threshold_i_short_square', 'threshold_t_long_square',
       'threshold_t_ramp', 'threshold_t_short_square',
       'threshold_v_long_square', 'threshold_v_ramp',
       'threshold_v_short_square', 'thumbnail_sweep_id',
       'trough_t_long_square', 'trough_t_ramp', 'trough_t_short_square',
       'trough_v_long_square', 'trough_v_ramp', 'trough_v_short_square',
       'upstroke_downstroke_ratio_long_square',
       'upstroke_downstroke_ratio_ramp',
       'upstroke_downstroke_ratio_short_square', 'vm_for_sag', 'vrest']

visal_ephys_data = visal_ephys_data.loc[:, ephys_list]
vispm_ephys_data = vispm_ephys_data.loc[:, ephys_list]


numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
newdf = mouse_ephys_df.select_dtypes(include=numerics)
newdf = newdf.drop(['structure_area_id', 'donor_id', 'normalized_depth',␣
 ↪'electrode_0_pa'], 1)
newdf = newdf.dropna(axis=0).dropna(axis=1)

visal_ephys_data = visal_ephys_data.dropna(axis=0).dropna(axis=1)
visal_ephys_num = visal_ephys_data.select_dtypes(include=numerics)
visal_ephys_name = visal_ephys_num.copy()
visal_column = np.repeat("VISal", visal_ephys_name.shape[0])
visal_ephys_name['Brain Region'] = visal_column

vispm_ephys_data = vispm_ephys_data.dropna(axis=0).dropna(axis=1)
vispm_ephys_num = vispm_ephys_data.select_dtypes(include=numerics)
```

```
vispm_ephys_name = vispm_ephys_num.copy()

vispm_column = np.repeat("VISpm", vispm_ephys_name.shape[0])
vispm_ephys_name['Brain Region'] = vispm_column

visalpm_ephys = pd.concat([visal_ephys_num,vispm_ephys_num])
visalpm_ephys_name = pd.concat([visal_ephys_name, vispm_ephys_name])
visalpm_ephys = (visalpm_ephys - visalpm_ephys.mean())/visalpm_ephys.std()
```

```
17
89
```

Related Columns list for further dissection in PCA analysis

[12]:
```
ephys_list_long_square = ['adaptation', 'avg_isi', 'f_i_curve_slope',
       'fast_trough_t_long_square', 'fast_trough_v_long_square',
       'input_resistance_mohm', 'latency',
       'peak_t_long_square',
       'threshold_i_long_square',
       'threshold_t_long_square',
       'threshold_v_long_square',
       'trough_t_long_square',
       'trough_v_long_square',
       'upstroke_downstroke_ratio_long_square',
       'vrest']


ephys_list_ramp =['adaptation', 'avg_isi', 'f_i_curve_slope',
        'fast_trough_t_ramp', 'fast_trough_v_ramp',
       'input_resistance_mohm', 'latency',
       'peak_t_ramp', 'peak_v_ramp',
       'threshold_i_ramp','threshold_t_ramp', 'threshold_v_ramp',
       'trough_t_ramp', 'trough_v_ramp',
       'upstroke_downstroke_ratio_ramp',
       'vrest']
```

6.2.3 Cre-lines distrubution of targeted cells in VISal and VISpm

[13]:
```
visal_CRE = pd.DataFrame(visal_ephys_transgenic['transgenic_line'].
 ↪value_counts())
visal_CRE.columns = ["Count"]

vispm_CRE = pd.DataFrame(vispm_ephys_transgenic['transgenic_line'].
 ↪value_counts())
vispm_CRE.columns = ["Count"]
```

6.2.4 Collecting related visual area data from Allen Brain Observatory

[14]:
```
print("all targeted structures: " + str(targeted_structures))

targeted_Cre = boc.get_all_cre_lines()
print("all Cre lines: " + str(targeted_Cre))

visal_data = boc.get_experiment_containers(targeted_structures=['VISal'])

visal_BO_cre = []
for items in visal_data:
    visal_BO_cre.append(items['cre_line'])

d = Counter(visal_BO_cre)
visal_BO_cre_df = pd.DataFrame.from_dict(d, orient='index')
print(d)
visal_BO_cre_df.columns = ['Count']

visal_id = visal_data[0]['id']
print("Num of experiments done in VISal: " + str(len(visal_data)))

vispm_data = boc.get_experiment_containers(targeted_structures=['VISpm'])

vispm_BO_cre = []
for items in vispm_data:
    vispm_BO_cre.append(items['cre_line'])

e = Counter(vispm_BO_cre)
vispm_BO_cre_df = pd.DataFrame.from_dict(e, orient='index')
print(e)
vispm_BO_cre_df.columns = ['Count']

vispm_id = vispm_data[0]['id']
print("Num of experiments done in VISpm: " + str(len(vispm_data)))

stim = boc.get_all_stimuli()
print("Different Scenes:")
print(stim)
```

```
all targeted structures: ['VISal', 'VISam', 'VISl', 'VISp', 'VISpm', 'VISrl']
all Cre lines: ['Cux2-CreERT2', 'Emx1-IRES-Cre', 'Fezf2-CreER', 'Nr5a1-Cre',
'Ntsr1-Cre_GN220', 'Pvalb-IRES-Cre', 'Rbp4-Cre_KL100', 'Rorb-IRES2-Cre',
'Scnn1a-Tg3-Cre', 'Slc17a7-IRES2-Cre', 'Sst-IRES-Cre', 'Tlx3-Cre_PL56', 'Vip-
IRES-Cre']
Counter({'Cux2-CreERT2': 13, 'Emx1-IRES-Cre': 7, 'Nr5a1-Cre': 6,
'Rbp4-Cre_KL100': 6, 'Rorb-IRES2-Cre': 6, 'Slc17a7-IRES2-Cre': 2, 'Sst-IRES-
Cre': 1})
```

```
Num of experiments done in VISal: 41
Counter({'Vip-IRES-Cre': 16, 'Slc17a7-IRES2-Cre': 15, 'Sst-IRES-Cre': 14,
'Cux2-CreERT2': 13, 'Rorb-IRES2-Cre': 7, 'Nr5a1-Cre': 7, 'Rbp4-Cre_KL100': 6,
'Ntsr1-Cre_GN220': 5, 'Emx1-IRES-Cre': 4})
Num of experiments done in VISpm: 87
Different Scenes:
['drifting_gratings', 'locally_sparse_noise', 'locally_sparse_noise_4deg',
'locally_sparse_noise_8deg', 'natural_movie_one', 'natural_movie_three',
'natural_movie_two', 'natural_scenes', 'spontaneous', 'static_gratings']
```

[15]:
```
# Download cells for a set of experiments and convert to DataFrame
cells = boc.get_cell_specimens()
cells = pd.DataFrame.from_records(cells)
print("Total cells: %d" % len(cells))

print("------------------------")
# find direction selective cells in VISal
visal_ec_ids = [ ec['id'] for ec in visal_data ]
visal_cells = cells[cells['experiment_container_id'].isin(visal_ec_ids)]
print("VISal cells: %d" % len(visal_cells))

# Response to drifting gratings stimulus
sig_al_cells = visal_cells[visal_cells['p_dg'] < 0.05]
print("Cells in anteriorlateral visual area with sig. response to drifting␣
 ↪gratings: %d" % len(sig_al_cells))

# Direction selective cells
dsi_al_cells = sig_al_cells[(sig_al_cells['g_dsi_dg'] > 0.9)]
print("Anteriorlateral visual area direction-selective cells: %d" %␣
 ↪len(dsi_al_cells))

print("------------------------")

# find direction selective cells in VISpm
vispm_ec_ids = [ ec['id'] for ec in vispm_data ]
vispm_cells = cells[cells['experiment_container_id'].isin(vispm_ec_ids)]
print("VISpm cells: %d" % len(vispm_cells))

# Response to drifting gratings stimulus
sig_pm_cells = vispm_cells[vispm_cells['p_dg'] < 0.05]
print("Cells in posteriorlateral visual area with sig. response to drifting␣
 ↪gratings: %d" % len(sig_pm_cells))

# Direction selective cells
dsi_pm_cells = sig_pm_cells[(sig_pm_cells['g_dsi_dg'] > 0.9)]
print("Posteriorlateral visual area direction-selective cells: %d" %␣
 ↪len(dsi_pm_cells))
```

```
dsi_data = {'Properties': ['Direction Selective', 'Total responding',␣
 ↪'Direction Selective', 'Total responding'],
            'Count': [len(dsi_al_cells), len(sig_al_cells) , len(dsi_pm_cells),␣
 ↪len(sig_pm_cells)],
            'Brain Region': ['VISal', 'VISal', 'VISpm', 'VISpm']}
al_pm_dsi = pd.DataFrame(dsi_data)
al_pm_sig = al_pm_dsi[al_pm_dsi['Properties'] == 'Total responding']
al_pm_d = al_pm_dsi[al_pm_dsi['Properties'] == 'Direction Selective']
```

```
Total cells: 63251
------------------------
VISal cells: 7191
Cells in anteriorlateral visual area with sig. response to drifting gratings:
3517
Anteriorlateral visual area direction-selective cells: 169
------------------------
VISpm cells: 7985
Cells in posteriorlateral visual area with sig. response to drifting gratings:
3220
Posteriorlateral visual area direction-selective cells: 171
```

[16]:
```
# find experiment containers for those cells
dsi_al_ids = dsi_al_cells['experiment_container_id'].unique()
print("total al dsi experiment containers: %d" % len(dsi_al_ids))

# Download the ophys experiments containing the drifting gratings stimulus for␣
 ↪VISal experiment containers
dsi_al_exps = boc.get_ophys_experiments(experiment_container_ids=dsi_al_ids,␣
 ↪stimuli=[stim_info.DRIFTING_GRATINGS])
print("VISal drifting gratings ophys experiments: %d" % len(dsi_al_exps))
```

```
total al dsi experiment containers: 30
VISal drifting gratings ophys experiments: 30
```

6.2.5 Handling Mouse connectivity Database

[17]:
```
# open up a list of all of the experiments
all_experiments = mcc.get_experiments(dataframe=True)
print("There are %d total experiments on mouse connectivity" %␣
 ↪len(all_experiments))

VISal_acronym_list = ['VISal', 'VISal1', 'VISal2/3', 'VISal4', 'VISal5',␣
 ↪'VISal6a', 'VISal6b']
VISal_acronym = structure_tree.get_structures_by_acronym(VISal_acronym_list)
```

```
VISpm_acronym_list = ['VISpm', 'VISpm1', 'VISpm2/3', 'VISpm4', 'VISpm5',␣
↪'VISpm6a', 'VISpm6b']
VISpm_acronym = structure_tree.get_structures_by_acronym(VISpm_acronym_list)
print("There are %d total experiments on VISpm connectivity" %␣
↪len(VISpm_acronym))
```

```
There are 2992 total experiments on mouse connectivity
There are 7 total experiments on VISpm connectivity
```

[18]:
```
summary_structures = structure_tree.get_structures_by_set_id([167587189])
pd.DataFrame(summary_structures)
print(len(summary_structures))
```

```
316
```

[19]:
```
# fetch the experiments that have injections in the isocortex of cre-positive␣
↪mice
isocortex = structure_tree.get_structures_by_name(['Isocortex'])[0]
cre_cortical_experiments = mcc.get_experiments(cre=True,␣
↪injection_structure_ids=[isocortex['id']])
print("%d cre cortical experiments" % len(cre_cortical_experiments))
```

```
1209 cre cortical experiments
```

[20]:
```
#find wild-type injections into AnterioLateral visual area
visal = structure_tree.get_structures_by_acronym(VISal_acronym_list)[0]
visal_experiments = mcc.get_experiments(cre=True,
                                        injection_structure_ids=[visal['id']])

print("%d VISal experiments" % len(visal_experiments))

al_structure_unionizes = mcc.get_structure_unionizes([ e['id'] for e in␣
↪visal_experiments ],
                                                     is_injection=False,
                                                    ␣
↪structure_ids=[isocortex['id']],
                                                     include_descendants=True)

print("%d VISal non-injection, cortical structure unionizes" %␣
↪len(al_structure_unionizes))
al_structure_unionizes.head()
```

```
17 VISal experiments
15045 VISal non-injection, cortical structure unionizes
```

[20]:

	experiment_id	hemisphere_id	id	is_injection	max_voxel_density	\
0	520996382	3	640783886	False	0.173927	

```
1      520996382              1  640782585         False           0.546264
2      520996382              3  640783961         False           0.676250
3      520996382              2  640777871         False           0.000000
4      520996382              3  640782721         False           0.813034

   max_voxel_x  max_voxel_y  max_voxel_z  normalized_projection_volume  \
0         6650         1220         3380                      0.001168
1         5170         4910         2290                      0.005493
2         3990         1490         7780                      0.051467
3            0            0            0                      0.000000
4         7970         1350         2040                      1.245962

   projection_density  projection_energy  projection_intensity  \
0            0.000322          0.199142            618.587911
1            0.000099          0.093984            949.327605
2            0.000323          0.197320            611.805389
3            0.000000          0.000000              0.000000
4            0.036146         33.858096            936.699871

   projection_volume  structure_id  sum_pixel_intensity    sum_pixels  \
0           0.000092     182305705         7.397989e+10  2.321802e+08
1           0.000431            95         7.746513e+11  3.552348e+09
2           0.004036           985         2.654921e+12  1.021620e+10
3           0.000000           201         2.135174e+11  6.822180e+08
4           0.097714            22         5.960447e+11  2.206769e+09

   sum_projection_pixel_intensity  sum_projection_pixels     volume
0                    4.623686e+07           7.474581e+04   0.284421
1                    3.338650e+08           3.516858e+05   4.351626
2                    2.015858e+09           3.294933e+06  12.514846
3                    0.000000e+00           0.000000e+00   0.835717
4                    7.471698e+10           7.976619e+07   2.703291
```

[21]:
```
dense_unionizes = al_structure_unionizes[ al_structure_unionizes.
 ↪projection_density > .1 ]
large_unionizes = dense_unionizes[ dense_unionizes.volume > .5 ]
large_structures = pd.DataFrame(structure_tree.nodes(large_unionizes.
 ↪structure_id))

print("%d large, dense, cortical, non-injection unionizes, %d structures" % (␣
 ↪len(large_unionizes), len(large_structures) ))

print(large_structures.name)

large_unionizes
```

```
69 large, dense, cortical, non-injection unionizes, 69 structures
```

```
0      Primary visual area, layer 4
1         Rostrolateral visual area
2     Primary visual area, layer 6a
3                     Anterior area
4          Anteromedial visual area
                   …
64    Primary visual area, layer 6a
65        posteromedial visual area
66     Primary visual area, layer 1
67                      Visual areas
68                  Postrhinal area
Name: name, Length: 69, dtype: object
```

[21]:

	experiment_id	hemisphere_id	id	is_injection	\
137	520996382	1	640781318	False	
447	520996382	1	640781888	False	
583	520996382	1	640782648	False	
2798	528509838	1	640901917	False	
3018	528509838	3	640903696	False	
…	…	…	…	…	
14466	524666904	1	640851737	False	
14530	524666904	1	640851340	False	
14547	524666904	1	640851295	False	
14583	524666904	1	640851227	False	
14656	524666904	1	640850120	False	

	max_voxel_density	max_voxel_x	max_voxel_y	max_voxel_z	\
137	0.729122	8780	990	3360	
447	0.813034	7970	1350	2040	
583	0.709272	8810	1480	3150	
2798	1.000000	7620	510	3530	
3018	1.000000	7730	530	3590	
…	…	…	…	…	
14466	0.898405	8480	1630	2570	
14530	1.000000	7950	570	3730	
14547	0.999606	8840	590	3290	
14583	1.000000	7950	570	3730	
14656	0.999871	9760	2730	1300	

	normalized_projection_volume	projection_density	projection_energy	\
137	0.844619	0.117386	151.556244	
447	0.807299	0.113947	112.433046	
583	0.813943	0.108134	117.561356	
2798	2.357872	0.111205	219.006930	
3018	2.877652	0.123223	291.729062	
…	…	…	…	
14466	1.573507	0.113063	110.957413	

```
14530                     2.381890            0.172917         236.683459
14547                     2.021766            0.125489         137.556625
14583                    18.312359            0.113560         133.211155
14656                     2.602081            0.154486         159.055633

       projection_intensity  projection_volume  structure_id  \
137              1291.096313          0.066239           721
447               986.715732          0.063312           417
583              1087.184326          0.063833            33
2798             1969.401458          0.088223     312782546
3018             2367.493549          0.107671           394
…                      …                 …             …
14466             981.377075          0.066041            33
14530            1368.769523          0.099969           533
14547            1096.167969          0.084854           593
14583            1173.048500          0.768575           669
14656            1029.578569          0.109210     312782628

       sum_pixel_intensity    sum_pixels  sum_projection_pixel_intensity  \
137           2.050862e+11  4.606380e+08                     6.981257e+10
447           1.633774e+11  4.535721e+08                     5.099649e+10
583           1.704421e+11  4.818888e+08                     5.665150e+10
2798          3.795801e+11  6.476238e+08                     1.418341e+11
3018          4.481023e+11  7.133022e+08                     2.080910e+11
…                    …             …                              …
14466         1.470833e+11  4.768200e+08                     5.290671e+10
14530         1.957105e+11  4.719438e+08                     1.117013e+11
14547         1.509399e+11  5.519916e+08                     7.593011e+10
14583         1.788725e+12  5.524917e+09                     7.359806e+11
14656         1.971878e+11  5.770818e+08                     9.178811e+10

       sum_projection_pixels    volume
137             5.407232e+07  0.564282
447             5.168306e+07  0.555626
583             5.210846e+07  0.590314
2798            7.201889e+07  0.793339
3018            8.789506e+07  0.873795
…                        …         …
14466           5.391069e+07  0.584105
14530           8.160709e+07  0.578131
14547           6.926868e+07  0.676190
14583           6.274085e+08  6.768024
14656           8.915115e+07  0.706925

[69 rows x 19 columns]
```

[22]:
```
# find wild-type injections into PosteriorMedial visual area
vispm = structure_tree.get_structures_by_acronym(['VISpm'])[0]
vispm_experiments = mcc.get_experiments(cre=True,
                                        injection_structure_ids=[vispm['id']])

print("%d VISpm experiments" % len(vispm_experiments))

pm_structure_unionizes = mcc.get_structure_unionizes([ e['id'] for e in␣
 ↪vispm_experiments ],
                                                     is_injection=False,
                                                    ␣
 ↪structure_ids=[isocortex['id']],
                                                     include_descendants=True)

print("%d VISpm non-injection, cortical structure unionizes" %␣
 ↪len(pm_structure_unionizes))

pm_structure_unionizes.head()
```

```
27 VISpm experiments
23895 VISpm non-injection, cortical structure unionizes
```

[22]:

	experiment_id	hemisphere_id	id	is_injection	max_voxel_density	\
0	523718823	3	640839159	False	0.465228	
1	523718823	1	640837833	False	0.000206	
2	523718823	1	640837916	False	0.354538	
3	523718823	3	640839140	False	0.235882	
4	523718823	1	640837801	False	0.522762	

	max_voxel_x	max_voxel_y	max_voxel_z	normalized_projection_volume	\
0	9150	2760	1240	4.484601e-02	
1	6590	1050	3510	6.393456e-09	
2	6000	650	4930	7.591707e-02	
3	4470	1570	5420	7.468156e-03	
4	8060	1930	1550	2.989156e-02	

	projection_density	projection_energy	projection_intensity	\
0	9.782284e-03	6.643982e+00	679.185155	
1	1.667878e-09	6.326772e-07	379.330627	
2	4.237331e-04	1.575204e-01	371.744303	
3	8.432550e-04	4.842790e-01	574.297151	
4	1.246139e-02	9.529721e+00	764.739929	

	projection_volume	structure_id	sum_pixel_intensity	sum_pixels	\
0	3.373732e-03	234	6.253321e+10	2.815362e+08	
1	4.809750e-10	577	7.551257e+10	2.354085e+08	
2	5.711185e-03	500	2.791571e+12	1.100266e+10	

```
3       5.618239e-04           211         1.418957e+11  5.438826e+08
4       2.248720e-03           600         2.948636e+10  1.473102e+08

   sum_projection_pixel_intensity  sum_projection_pixels     volume
0                     1.870521e+09           2.754067e+06   0.344882
1                     1.489376e+02           3.926327e-01   0.288375
2                     1.733143e+09           4.662192e+06  13.478262
3                     2.633909e+08           4.586318e+05   0.666256
4                     1.403825e+09           1.835690e+06   0.180455
```

[23]:
```
dense_unionizes = pm_structure_unionizes[ pm_structure_unionizes.
 ↪projection_density > .1 ]
large_unionizes = dense_unionizes[ dense_unionizes.volume > .5 ]
large_structures = pd.DataFrame(structure_tree.nodes(large_unionizes.
 ↪structure_id))

print("%d large, dense, cortical, non-injection unionizes, %d structures" % (␣
 ↪len(large_unionizes), len(large_structures) ))

print(large_structures.name)

large_unionizes
```

```
141 large, dense, cortical, non-injection unionizes, 141 structures
0                                    Anterior area
1                   Primary visual area, layer 6a
2                             Lateral visual area
3                  Primary visual area, layer 2/3
4                    Primary visual area, layer 1
                           …
136                 Primary visual area, layer 6a
137                  Primary visual area, layer 1
138                      Anteromedial visual area
139    Retrosplenial area, lateral agranular part
140                 Primary visual area, layer 6a
Name: name, Length: 141, dtype: object
```

[23]:
```
       experiment_id  hemisphere_id         id  is_injection  \
962        294481346              2  632547103         False
1063       294481346              2  632544815         False
1160       294481346              2  632545607         False
1209       294481346              2  632546768         False
1261       294481346              2  632546003         False
…              …              …         …             …
20986      590987294              1  641365079         False
21636      485237081              1  640503011         False
21902      485237081              3  640504326         False
```

```
21951      485237081              1  640502776         False
22067      485237081              1  640503828         False

       max_voxel_density  max_voxel_x  max_voxel_y  max_voxel_z  \
962             0.999823         7650          840         7870
1063            1.000000         8700         1130         7620
1160            0.998055         8740         1840         9460
1209            1.000000         8780          700         7660
1261            1.000000         8700          430         7810
…                  …            …            …           …
20986           0.998225         8630         1030         4010
21636           0.999965         9950          860         3730
21902           0.939685         7630          390         4100
21951           0.999011         8710          910         4300
22067           1.000000         8480         1150         3810

       normalized_projection_volume  projection_density  projection_energy  \
962                        0.769835            0.126034          81.467639
1063                       0.978650            0.217842         173.360535
1160                       0.563655            0.108357          60.055810
1209                       1.237950            0.146156         172.762436
1261                       0.858260            0.160055         185.093109
…                             …                 …                 …
20986                      0.548545            0.114227         100.877289
21636                      0.836540            0.113053         162.015076
21902                      1.233929            0.125849         134.986099
21951                      1.635542            0.116556          87.194175
22067                      0.857586            0.140019         152.334869

       projection_intensity  projection_volume  structure_id  \
962              646.392702           0.099988     312782546
1063             795.806885           0.127109            33
1160             554.240007           0.073209           409
1209            1182.038818           0.160788           821
1261            1156.436035           0.111473           593
…                     …                …             …
20986            883.128479           0.064384            33
21636           1433.089600           0.074477           593
21902           1072.604878           0.109857           394
21951            748.089901           0.145613           894
22067           1087.958740           0.076351            33

       sum_pixel_intensity    sum_pixels  sum_projection_pixel_intensity  \
962           1.235053e+11  6.476238e+08                    5.276038e+10
1063          1.363023e+11  4.763197e+08                    8.257505e+10
1160          9.828260e+10  5.515308e+08                    3.312263e+10
1209          2.565085e+11  8.980472e+08                    1.551488e+11
```

```
1261          1.514309e+11  5.685435e+08                    1.052335e+11
…                      …           …                             …
20986         1.542229e+11  4.601245e+08                    4.641611e+10
21636         1.475572e+11  5.377815e+08                    8.712871e+10
21902         2.337040e+11  7.125939e+08                    9.619027e+10
21951         2.517442e+11  1.019834e+09                    8.892358e+10
22067         1.514811e+11  4.451355e+08                    6.780966e+10

       sum_projection_pixels    volume
962             8.162280e+07  0.793339
1063            1.037627e+08  0.583492
1160            5.976225e+07  0.675625
1209            1.312553e+08  1.100108
1261            9.099810e+07  0.696466
…                        …       …
20986           5.255873e+07  0.563652
21636           6.079781e+07  0.658782
21902           8.967913e+07  0.872928
21951           1.188675e+08  1.249297
22067           6.232742e+07  0.545291

[141 rows x 19 columns]
```

6.3 Data Analysis & Results

Include cells that describe the steps in your data analysis.

6.3.1 1.1 Plotting compartive data between the genetic fold change of VISal and VISpm

[39]:
```
# plotting the gene expression data through barplot
fig, ax = plt.subplots(ncols = 2,figsize=(10,6))
NL_VISal_plot= NL_VISal_pd_fold[NL_VISal_pd_fold["fold-change"] > 2.7].copy()
NL_VISpm_plot = NL_VISpm_pd_fold[NL_VISpm_pd_fold["fold-change"] > 2].copy()

print("Num of gene plotted for VISal: " + str(len(NL_VISal_plot)))
print("Num of gene plotted for VISpm: " + str(len(NL_VISpm_plot)))

ax1 = sns.barplot(x = "gene-symbol", y = "fold-change", data = NL_VISal_plot,␣
↪ax = ax[0], color='#013ADF')
ax1.set_xticklabels(labels = NL_VISal_plot['gene-symbol'], rotation=90)
ax1.set_ylim(2.7,)
ax1.set_title('Gene Expression of VISal area comparing to Primary Visual␣
↪Cortex')
```

```
custom_palette = sns.color_palette("RdGy", 2)
ax2 = sns.barplot(x = "gene-symbol", y = "fold-change", data = NL_VISpm_plot,␣
↪ax = ax[1], color='#FE2E2E' )
ax2.set_xticklabels(labels = NL_VISpm_plot['gene-symbol'], rotation=90)
ax2.set_ylim(2,)
ax2.set_title('Gene Expression of VISpm area comparing to Primary Visual␣
↪Cortex')
```

```
Num of gene plotted for VISal: 56
Num of gene plotted for VISpm: 54
```

[39]: `Text(0.5, 1.0, 'Gene Expression of VISpm area comparing to Primary Visual Cortex')`

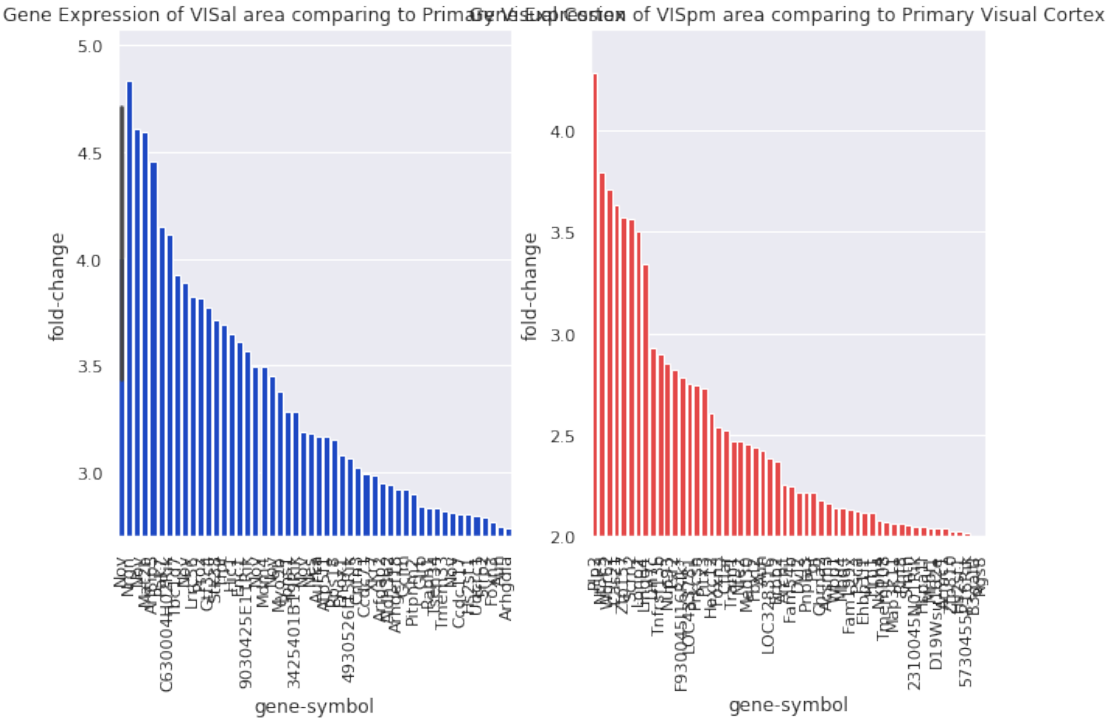


Figure 1: The plot on the left shows the two extremes of gene expression from RNA sequence in the anterolateral area (AL) compared to the posteromedial area (PM). The plot on the left compared gene expression posteromedial area (PM) to the anterolateral area (AL).

6.3.2 1.2 Further catergorizing the differential functional differences between VISal and VISPm using Gene Ontology

Isolate Entrez of all the genes identified in the ISH data

[25]:
```
NL_VISal_pd.sort_values(by=['fold-change'], ascending = False)
NL_VISpm_pd.sort_values(by=['fold-change'], ascending = False)
entrez_nlal = pd.DataFrame(NL_VISal_pd[['entrez-id']])
entrez_nlpm = pd.DataFrame(NL_VISpm_pd[['entrez-id']])
```

Export entrez ID to .csv format, for further analysis

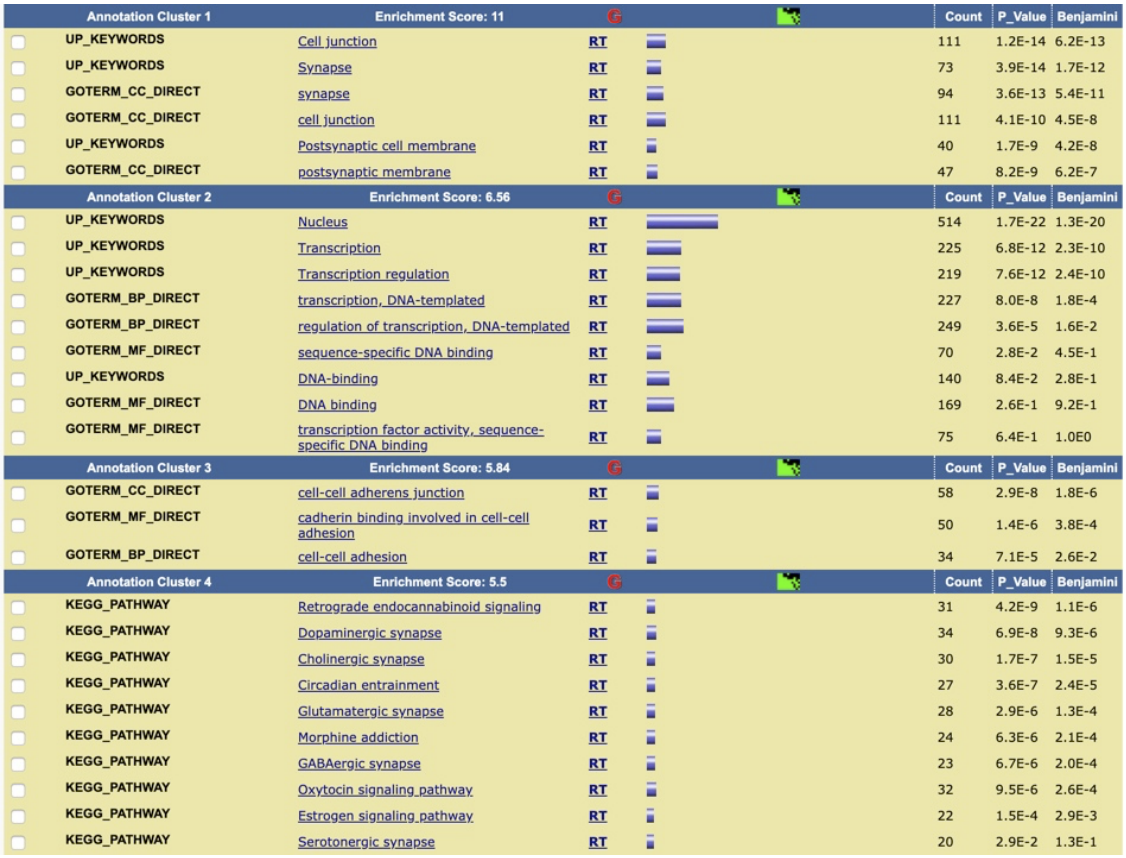
[26]:
```
entrez_nlal.to_csv("./entrez_nlal.csv", sep=',',index=False)
entrez_nlpm.to_csv("./entrez_nlpm.csv", sep=',',index=False)
```

6.4 DAVID analysis

6.4.1 Gene ontology result for Posterior Medial area comparing to primary visual cortex

Annotation Cluster 1	Enrichment Score: 9.16			Count	P_Value	Benjamini
GOTERM_CC_DIRECT	synapse	RT		89	8.2E-12	2.1E-9
UP_KEYWORDS	Synapse	RT		64	2.0E-10	7.7E-9
UP_KEYWORDS	Cell junction	RT		96	8.4E-10	3.0E-8
GOTERM_CC_DIRECT	cell junction	RT		100	1.8E-7	1.0E-5
Annotation Cluster 2	**Enrichment Score: 7.79**			**Count**	**P_Value**	**Benjamini**
UP_KEYWORDS	Nucleus	RT		500	1.1E-21	1.5E-19
UP_KEYWORDS	Transcription regulation	RT		221	1.8E-13	1.2E-11
UP_KEYWORDS	Transcription	RT		223	1.7E-12	8.3E-11
GOTERM_BP_DIRECT	transcription, DNA-templated	RT		226	1.7E-8	7.8E-5
GOTERM_BP_DIRECT	regulation of transcription, DNA-templated	RT		257	3.4E-7	7.9E-4
UP_KEYWORDS	DNA-binding	RT		156	7.4E-4	9.0E-3
GOTERM_MF_DIRECT	sequence-specific DNA binding	RT		76	1.1E-3	6.1E-2
GOTERM_MF_DIRECT	DNA binding	RT		183	6.8E-3	2.0E-1
GOTERM_MF_DIRECT	transcription factor activity, sequence-specific DNA binding	RT		95	6.9E-3	2.0E-1
Annotation Cluster 3	**Enrichment Score: 6.23**			**Count**	**P_Value**	**Benjamini**
UP_KEYWORDS	Synapse	RT		64	2.0E-10	7.7E-9
GOTERM_CC_DIRECT	postsynaptic density	RT		47	3.9E-8	2.7E-6
UP_KEYWORDS	Postsynaptic cell membrane	RT		30	5.5E-5	8.6E-4
GOTERM_CC_DIRECT	postsynaptic membrane	RT		35	2.9E-4	8.4E-3
Annotation Cluster 4	**Enrichment Score: 5.74**			**Count**	**P_Value**	**Benjamini**
GOTERM_CC_DIRECT	Golgi apparatus	RT		147	3.8E-7	1.7E-5
UP_KEYWORDS	Golgi apparatus	RT		95	9.2E-7	2.1E-5
GOTERM_CC_DIRECT	Golgi membrane	RT		59	1.8E-5	6.7E-4
Annotation Cluster 5	**Enrichment Score: 4.11**			**Count**	**P_Value**	**Benjamini**
KEGG_PATHWAY	Amphetamine addiction	RT		19	8.0E-6	1.1E-3
KEGG_PATHWAY	cAMP signaling pathway	RT		37	8.5E-6	7.8E-4
KEGG_PATHWAY	Dopaminergic synapse	RT		27	5.3E-5	2.9E-3
KEGG_PATHWAY	Circadian entrainment	RT		21	1.9E-4	6.4E-3
KEGG_PATHWAY	Oxytocin signaling pathway	RT		24	4.1E-3	6.4E-2

The image above is the result of DAVID analysis on the VISpm region. The top three clusters contain genes that are essential genes that all neurons have, they are not significant for the functional specialization.



Annotation Cluster 1	Enrichment Score: 11			Count	P_Value	Benjamini
UP_KEYWORDS	Cell junction	RT		111	1.2E-14	6.2E-13
UP_KEYWORDS	Synapse	RT		73	3.9E-14	1.7E-12
GOTERM_CC_DIRECT	synapse	RT		94	3.6E-13	5.4E-11
GOTERM_CC_DIRECT	cell junction	RT		111	4.1E-10	4.5E-8
UP_KEYWORDS	Postsynaptic cell membrane	RT		40	1.7E-9	4.2E-8
GOTERM_CC_DIRECT	postsynaptic membrane	RT		47	8.2E-9	6.2E-7
Annotation Cluster 2	**Enrichment Score: 6.56**			**Count**	**P_Value**	**Benjamini**
UP_KEYWORDS	Nucleus	RT		514	1.7E-22	1.3E-20
UP_KEYWORDS	Transcription	RT		225	6.8E-12	2.3E-10
UP_KEYWORDS	Transcription regulation	RT		219	7.6E-12	2.4E-10
GOTERM_BP_DIRECT	transcription, DNA-templated	RT		227	8.0E-8	1.8E-4
GOTERM_BP_DIRECT	regulation of transcription, DNA-templated	RT		249	3.6E-5	1.6E-2
GOTERM_MF_DIRECT	sequence-specific DNA binding	RT		70	2.8E-2	4.5E-1
UP_KEYWORDS	DNA-binding	RT		140	8.4E-2	2.8E-1
GOTERM_MF_DIRECT	DNA binding	RT		169	2.6E-1	9.2E-1
GOTERM_MF_DIRECT	transcription factor activity, sequence-specific DNA binding	RT		75	6.4E-1	1.0E0
Annotation Cluster 3	**Enrichment Score: 5.84**			**Count**	**P_Value**	**Benjamini**
GOTERM_CC_DIRECT	cell-cell adherens junction	RT		58	2.9E-8	1.8E-6
GOTERM_MF_DIRECT	cadherin binding involved in cell-cell adhesion	RT		50	1.4E-6	3.8E-4
GOTERM_BP_DIRECT	cell-cell adhesion	RT		34	7.1E-5	2.6E-2
Annotation Cluster 4	**Enrichment Score: 5.5**			**Count**	**P_Value**	**Benjamini**
KEGG_PATHWAY	Retrograde endocannabinoid signaling	RT		31	4.2E-9	1.1E-6
KEGG_PATHWAY	Dopaminergic synapse	RT		34	6.9E-8	9.3E-6
KEGG_PATHWAY	Cholinergic synapse	RT		30	1.7E-7	1.5E-5
KEGG_PATHWAY	Circadian entrainment	RT		27	3.6E-7	2.4E-5
KEGG_PATHWAY	Glutamatergic synapse	RT		28	2.9E-6	1.3E-4
KEGG_PATHWAY	Morphine addiction	RT		24	6.3E-6	2.1E-4
KEGG_PATHWAY	GABAergic synapse	RT		23	6.7E-6	2.0E-4
KEGG_PATHWAY	Oxytocin signaling pathway	RT		32	9.5E-6	2.6E-4
KEGG_PATHWAY	Estrogen signaling pathway	RT		22	1.5E-4	2.9E-3
KEGG_PATHWAY	Serotonergic synapse	RT		20	2.9E-2	1.3E-1

The image above is the result of DAVID analysis on the VISal region. The top three clusters contain genes that are essential genes that all neurons have, they are not significant for the functional specialization.

6.4.2 Cell Types Database: plotting resting potential of neurons VISal and VISpm areas

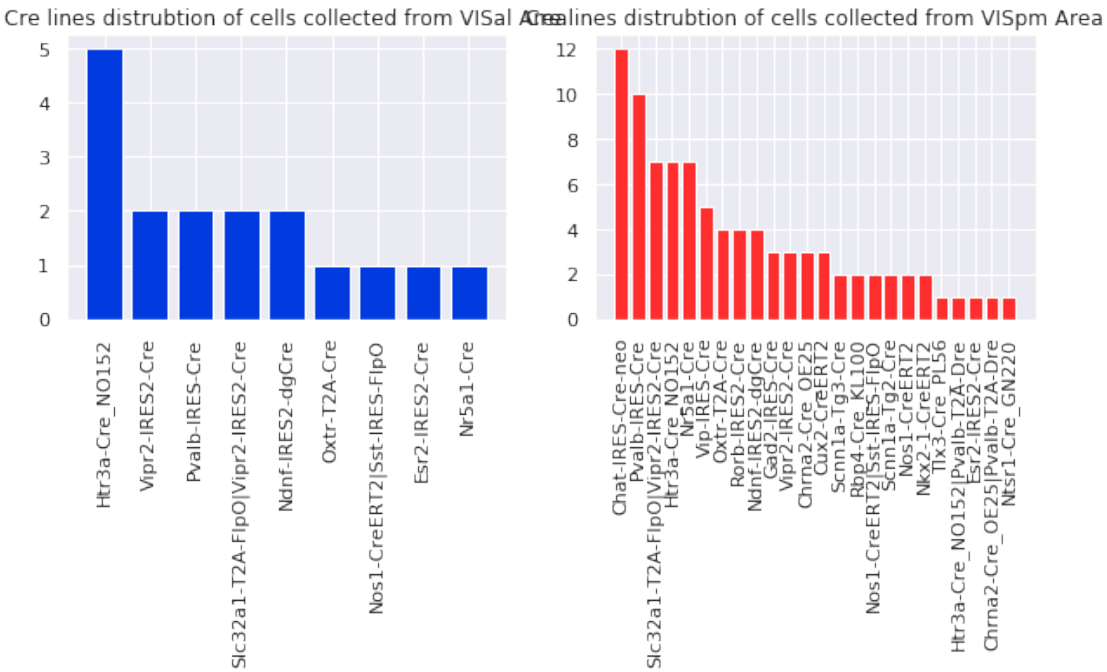
6.4.3 Verification: Cre-lines distrubution of the recorded VISal and VISpm cells

[27]:
```
fig, ax0 = plt.subplots(ncols = 2, figsize=(10,3))

ax0[0].bar(visal_CRE.index, visal_CRE['Count'], color = "#013ADF")
ax0[0].set_title('Cre lines distrubtion of cells collected from VISal Area')
ax0[0].set_xticklabels(labels = visal_CRE.index, rotation=90)
ax0[1].bar(vispm_CRE.index, vispm_CRE['Count'], color = "#FE2E2E")
ax0[1].set_title('Cre lines distrubtion of cells collected from VISpm Area')
ax0[1].set_xticklabels(labels = vispm_CRE.index, rotation=90)
plt.xticks(rotation=90)
```

[27]:
```
([0,
  1,
```

```
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23],
<a list of 24 Text xticklabel objects>)
```



[28]:
```
fig, ax1 = plt.subplots(ncols = 1, figsize=(12,8))
custom = ["#013ADF","#FE2E2E"]

sns.violinplot(y = 'Vrest', x = 'Brain Region', data = vrest, ax = ax1, palette␣
↪= custom, orient = 'v')
ax1.set_title('Resting Membrane Potential of VISal neurons comparing to VISpm␣
↪neurons')
ax1.set_ylabel('Resting Membrane Potential')
```

[28]:
```
Text(0, 0.5, 'Resting Membrane Potential')
```

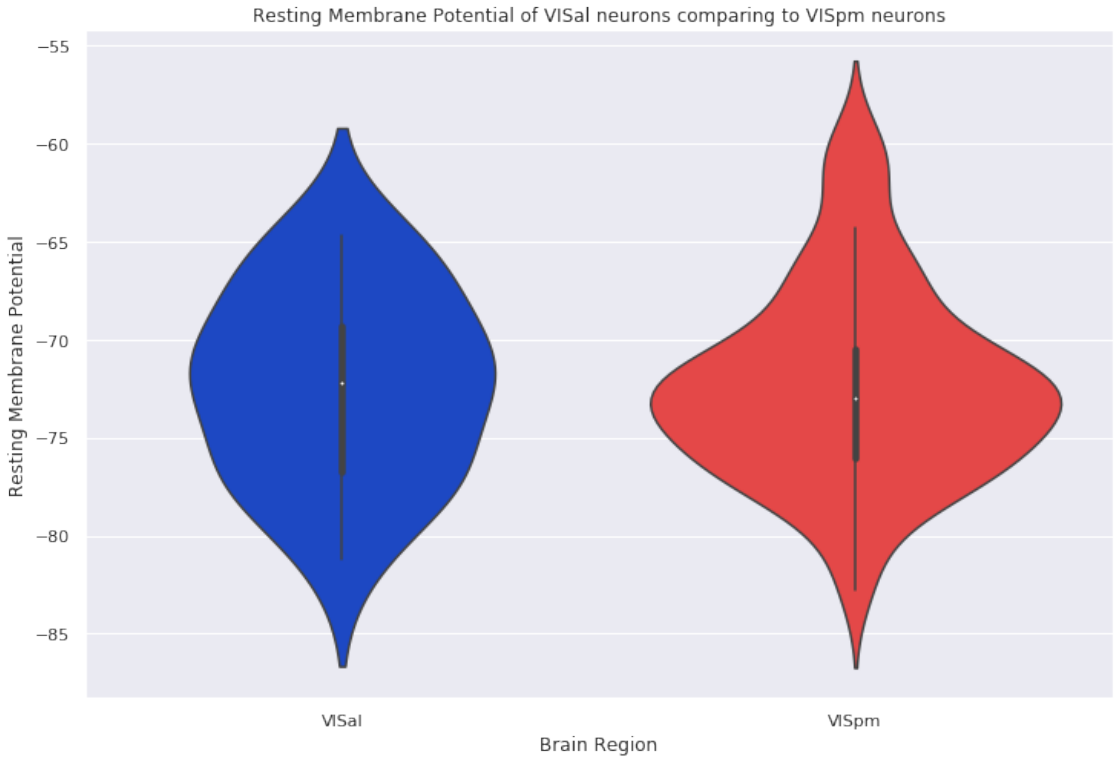


Figure 2: There are similarities in the range between the resting membrane potentials. The anterolateral area neurons' interquartile range ranges from -66 mV to -77 mV, while the resting membrane potential of the posteromedial area neuron's is -70 mV to -77 mV.

6.4.4 Running T-Test for the resting potential

[29]:
```
stats.ttest_ind(visal_ephys_vrest['Vrest'],vispm_ephys_vrest['Vrest'])
```

[29]:
```
Ttest_indResult(statistic=0.013586502817074395, pvalue=0.9891859021720485)
```

6.4.5 Looking into the electrophysiology data through PCA

[30]:
```
visalpm_cov = visalpm_ephys.cov()
plt.imshow(visalpm_cov)
plt.colorbar()
```

[30]:
```
<matplotlib.colorbar.Colorbar at 0x7f0bc30e7c88>
```

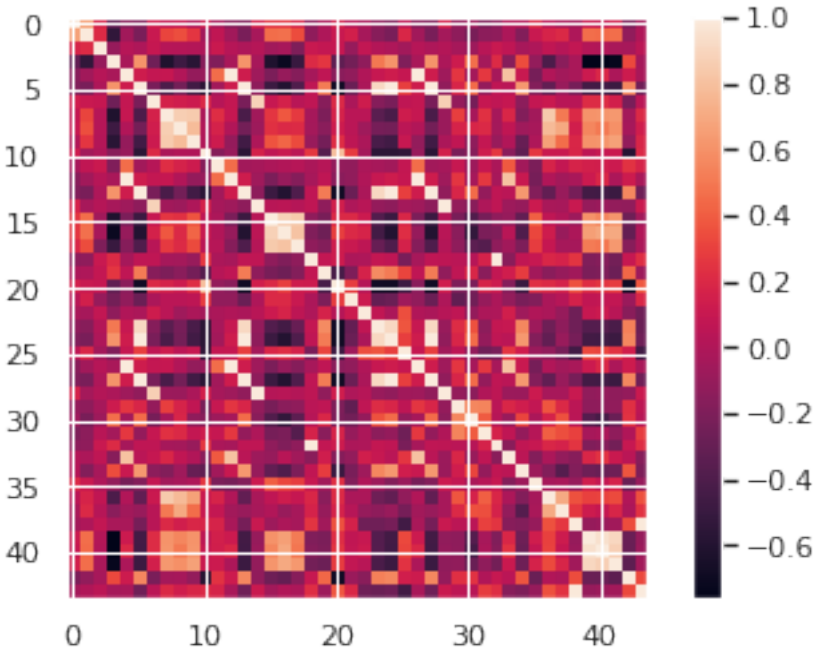


Figure 3 The covariance matrix shows there might be coorelations between electrophysiological features of VISal areas and VISpm areas. It will be further analysed through dimensionality reduction.

[31]:
```
pca = PCA(n_components = 5)
X_2D = pca.fit_transform(visalpm_ephys)
visalpm_ephys_name['PC1'] = X_2D[:,0]
visalpm_ephys_name['PC2'] = X_2D[:,1]
visalpm_ephys_name['PC3'] = X_2D[:,2]
visalpm_ephys_name['PC4'] = X_2D[:,3]
visalpm_ephys_name['PC5'] = X_2D[:,4]

X_2D_ls = pca.fit_transform(visalpm_ephys.loc[:, ephys_list_long_square])
visalpm_ephys_name['ls_PC1'] = X_2D_ls[:,0]
visalpm_ephys_name['ls_PC2'] = X_2D_ls[:,1]
visalpm_ephys_name['ls_PC3'] = X_2D_ls[:,2]
visalpm_ephys_name['ls_PC4'] = X_2D_ls[:,3]
```

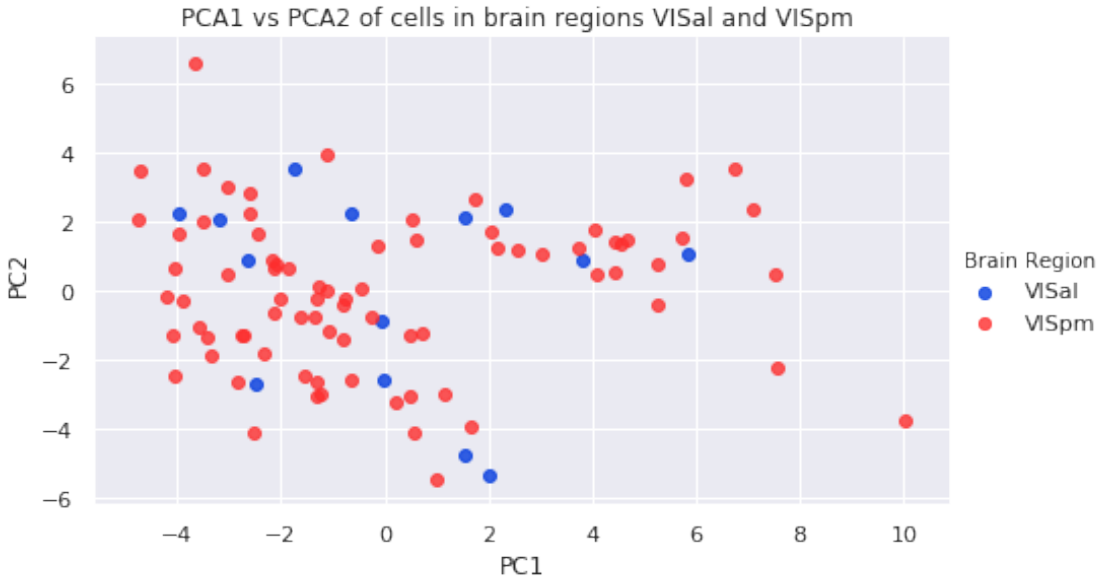
```
visalpm_ephys_name['ls_PC5'] = X_2D_ls[:,4]

X_2D_ramp = pca.fit_transform(visalpm_ephys.loc[:, ephys_list_ramp])
visalpm_ephys_name['r_PC1'] = X_2D_ramp[:,0]
visalpm_ephys_name['r_PC2'] = X_2D_ramp[:,1]
visalpm_ephys_name['r_PC3'] = X_2D_ramp[:,2]
visalpm_ephys_name['r_PC4'] = X_2D_ramp[:,3]
visalpm_ephys_name['r_PC5'] = X_2D_ramp[:,4]
```

[32]:
```
sns.lmplot("PC1", "PC2", hue ='Brain Region', palette = custom,␣
 ↪data=visalpm_ephys_name, fit_reg=False, height = 4, aspect = 1.7)
ax = plt.gca()
ax.set_title("PCA1 vs PCA2 of cells in brain regions VISal and VISpm ")

sns.lmplot("PC1", "PC3", hue ='Brain Region', palette = custom,␣
 ↪data=visalpm_ephys_name, fit_reg=False, height = 4, aspect = 1.7)
ax = plt.gca()
ax.set_title("PCA1 vs PCA3 of cells in brain regions VISal and VISpm ")
```

[32]:
```
Text(0.5, 1, 'PCA1 vs PCA3 of cells in brain regions VISal and VISpm ')
```



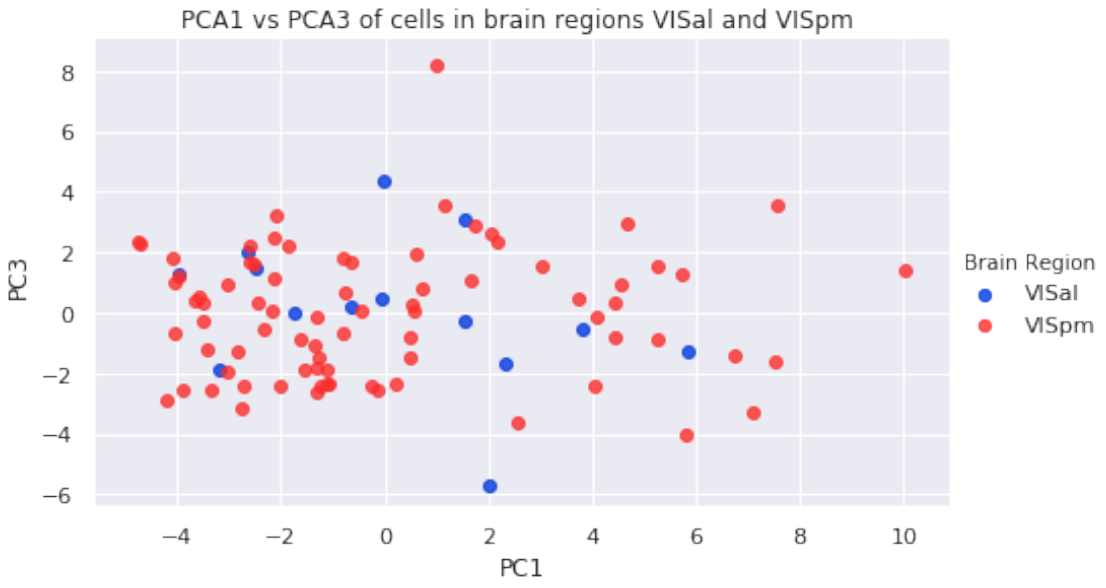
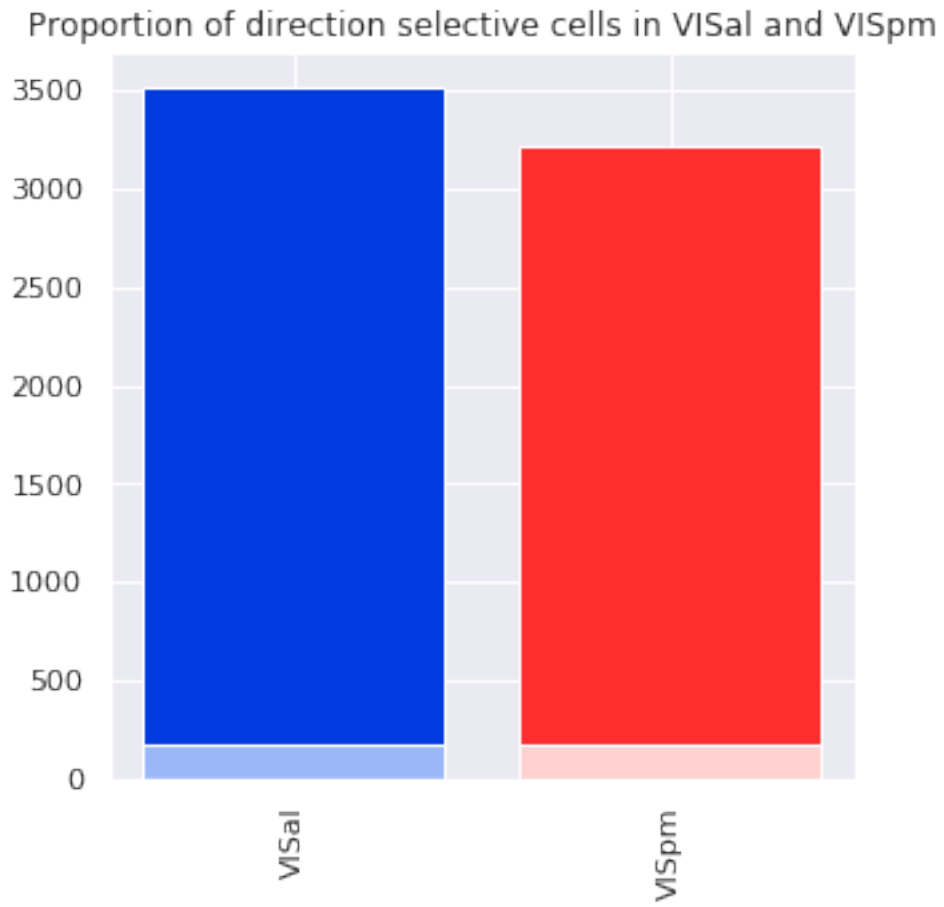


Figure 4 The principal components of combined electrophysiological(PCA) of AL and PM and compared to the stimulus types long square (PCA2) and ramp (PCA3) are plotted. There is no clear clustering of either variable showing no type of correlation between VISal and VISpm electrophysiology features

[33]:
```
plt.figure(figsize = (5,5))


plt.bar(al_pm_sig['Brain Region'], al_pm_sig['Count'], color = ['#013ADF',␣
↪'#FE2E2E'])
plt.bar(al_pm_d['Brain Region'], al_pm_d['Count'], color = ['#9ab8f9',␣
↪'#ffd1d1'])
plt.xticks(rotation=90)
plt.title("Proportion of direction selective cells in VISal and VISpm")
plt.show()
```



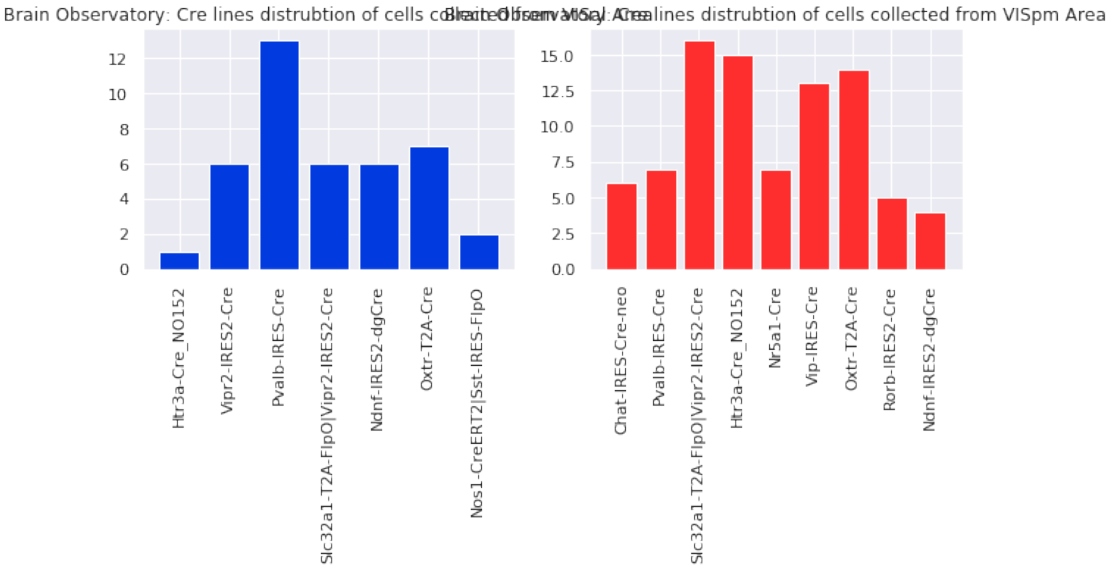
The plot above depicts the number of cells that respond to direction. The red bar represents the VISpm region while the blue bar depicts the VISal region.

6.4.6 Plots for Direction selective cells

[34]:
```
fig, ax3 = plt.subplots(ncols = 2, figsize=(10,3))

ax3[0].bar(visal_BO_cre_df.index, visal_BO_cre_df['Count'], color = "#013ADF")
ax3[0].set_title('Brain Observatory: Cre lines distrubtion of cells collected␣
 ↪from VISal Area')
ax3[0].set_xticklabels(labels = visal_CRE.index, rotation=90)
ax3[1].bar(vispm_BO_cre_df.index, vispm_BO_cre_df['Count'], color = "#FE2E2E")
ax3[1].set_title('Brain Observatory: Cre lines distrubtion of cells collected␣
 ↪from VISpm Area')
ax3[1].set_xticklabels(labels = vispm_CRE.index, rotation=90)
plt.xticks(rotation=90)
```

[34]: ([0, 1, 2, 3, 4, 5, 6, 7, 8], <a list of 9 Text xticklabel objects>)



6.4.7 Connectivity Database: Projection Matrix of VISal cells and VISpm cells

[35]:
```
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline

visal_experiment_ids = [ e['id'] for e in visal_experiments ]
ctx_children = structure_tree.child_ids( [isocortex['id']] )[0]

pm = mcc.get_projection_matrix(experiment_ids = visal_experiment_ids,
                               projection_structure_ids = ctx_children,
                               hemisphere_ids= [2], # right hemisphere, ipsilateral
                               parameter = 'projection_density')

row_labels = pm['rows'] # these are just experiment ids
column_labels = [ c['label'] for c in pm['columns'] ]
matrix = pm['matrix']

fig, ax = plt.subplots(figsize=(10,5))
heatmap = ax.pcolor(matrix, cmap=plt.cm.afmhot)

# put the major ticks at the middle of each cell
ax.set_xticks(np.arange(matrix.shape[1])+0.5, minor=False)
ax.set_yticks(np.arange(matrix.shape[0])+0.5, minor=False)
```

```
ax.set_xlim([0, matrix.shape[1]])
ax.set_ylim([0, matrix.shape[0]])

# want a more natural, table-like display
ax.set_xlabel('Brain Regions Acronyms', fontsize=16)
ax.xaxis.set_label_position('top')
ax.set_ylabel('Experiment Number', fontsize=16)
ax.yaxis.set_label_position('left')
ax.invert_yaxis()
ax.xaxis.tick_top()

ax.set_xticklabels(column_labels, minor=False)
ax.set_yticklabels(row_labels, minor=False)
fig.suptitle('Projection Matrix of VISal area', fontsize=20)
plt.show()
```

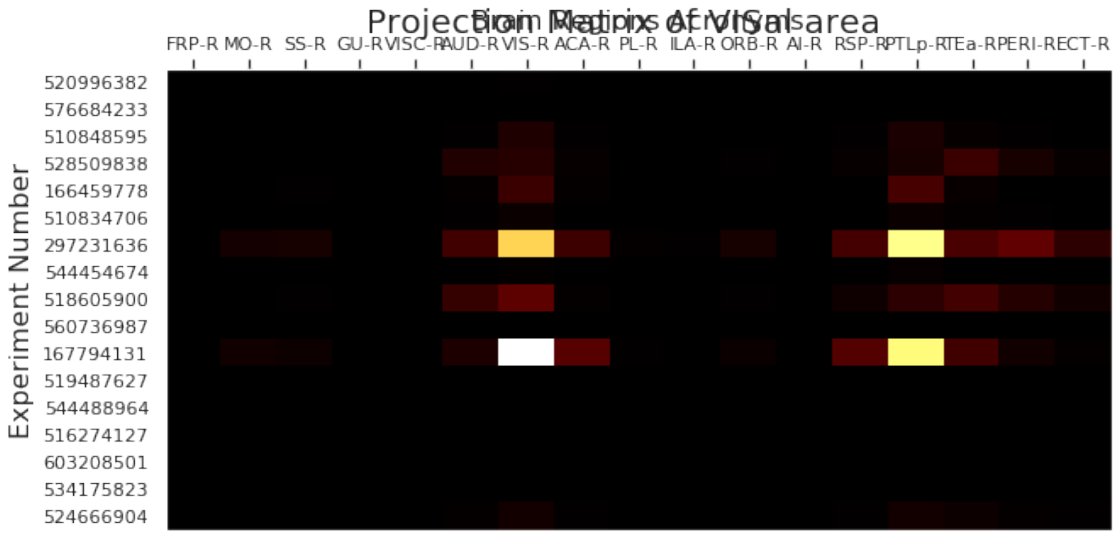


Figure 5 The figure above is a logarithmic heatmap of the projection density of the VISal region towards other brain structures. The projection density is very high between VISal region and the PTLp reg and between the VISal and the VIS region. The projection density is visually represented in the color of the heatmap with lighter colors indicating a stronger projection density.

[36]:
```
warnings.filterwarnings('ignore')
%matplotlib inline

vispm_experiment_ids = [ e['id'] for e in vispm_experiments ]
ctx_children = structure_tree.child_ids( [isocortex['id']] )[0]

pm = mcc.get_projection_matrix(experiment_ids = vispm_experiment_ids,
```

```
                                projection_structure_ids = ctx_children,
                                hemisphere_ids= [2], # right hemisphere,␣
↪ipsilateral
                                parameter = 'projection_density')

row_labels = pm['rows'] # these are just experiment ids
column_labels = [ c['label'] for c in pm['columns'] ]
matrix = pm['matrix']

fig, ax = plt.subplots(figsize=(10,5))
heatmap = ax.pcolor(matrix, cmap=plt.cm.afmhot)

# put the major ticks at the middle of each cell
ax.set_xticks(np.arange(matrix.shape[1])+0.5, minor=False)
ax.set_yticks(np.arange(matrix.shape[0])+0.5, minor=False)

ax.set_xlim([0, matrix.shape[1]])
ax.set_ylim([0, matrix.shape[0]])

# want a more natural, table-like display
ax.set_xlabel('Brain Regions Acronyms', fontsize=16)
ax.xaxis.set_label_position('top')
ax.set_ylabel('Experiment Number', fontsize=16)
ax.yaxis.set_label_position('left')
ax.invert_yaxis()
ax.xaxis.tick_top()

ax.set_xticklabels(column_labels, minor=False)
ax.set_yticklabels(row_labels, minor=False)
fig.suptitle('Projection Matrix of VISpm area', fontsize=20)

plt.show()
```

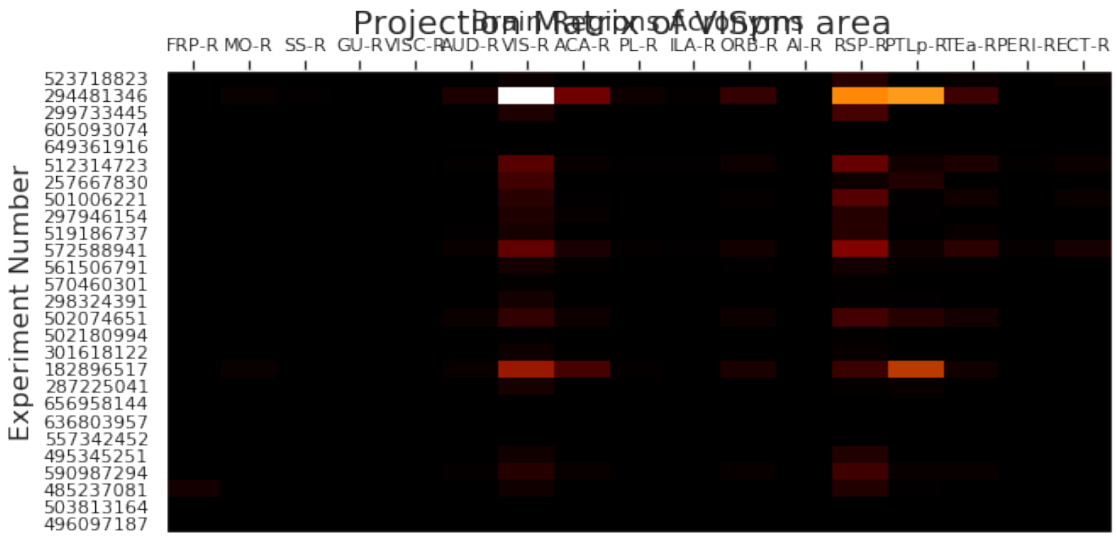


Figure 6 The figure above is a logarithmic heatmap of the projection density of the VISpm region towards other brain structures. The projection density is very high between VISal region and the RSP region and between the VISpm and the VIS region. The projection density is visually represented in the color of the heatmap with lighter colors indicating a stronger projection density.

6.5 Conclusion & Discussion

When comparing the resting membrane potential of the VISal and VISpm areas, we find them to be similar. The anterolateral area neurons have a median resting membrane potential of -71mV, while the median resting membrane potential of the posteromedial area neurons are -72. Although the VISal and VISpm are functionally specialized in two different areas, they are both located within the visual cortex of the brain which is likely why the resting membrane potentials are so similar. The functional specialization of the neurons has not impacted the resting potentials of the neurons very much.

Moreover, the anterolateral area and the posteromedial area of the visual cortex have a similar number of neurons that have a significant response to drifting gratings and a similar number of neurons that are direction selective. There are 7191 cells in the anterolateral area with 48.9% of them having a significant response to drifting gratings and 2.3% of them are direction selective. On the other hand, there are 7985 cells in the posterolateral visual area with 40.3% of them having a significant response to drifting gratings and 2.1% if them are direction selective.

Using PCA analysis, the figure above shows that there are samples correlated to the electrophysiological properties. Samples that are correlated will cluster together apart from samples that are not correlated. These electrophysiological measurements(long square and ramp) were selected as they represent both the passive and active membrane properties of the cell populations, while not including several measurements that could be correlated. Together with the data from resting membrane potential and the PCA, there are no differences between the electrophysiological properties. Huh et al also found that there are no differences between the electrophysiological properties between

the two brain regions.It is important to note that the Allen brain institute database does not have data for cells recording under different spatial frequency.

Looking at the directional selective, there more directional selective cells in the AL compared to the PM cells. Huh et al says AL L5 neurons were more strongly modulated by the phase of gratings and by spontaneous running of the animal compared to PM L5 neurons.[6] When handling visual stimuli, the two regions might have the same function handling direction but have differences provide the process in the speed of the object.

Upon analyzing the projection matrix of the VISal area, it can be observed that the projection density of the VISal region towards the PTLp region (posterior parietal association area) is much greater than the projection density of VISal to any other region. The only exception to this would be the fact that the VISal area has very high projection density towards the VIS region, however this is something to be expected as the VISal region is within the VIS region as depicted in the Allen Institute's Brain Atlas, therefore high projection density between the VISal region and the rest of the VIS region is to be expected. Our findings that the VISal region has high projection density towards the PTLp region resonates with the findings of a 2019 study (Lyamzin et. al, 2019) in which it was also observed that the PTLp region receives dense projections from the VISal region. The posterior parietal association area, which is also commonly referred to as the posterior parietal cortex, is an association area that brings together sensory information in order to form a cognitive representation of space (Anderson, 1997). A possible reasoning supported by the literature as to why we observed a very high projection density between the VISal region and the PLTp region may be that the VISal region takes visual stimuli to the PTLp region which then is tasked with constructing a representation of the space surrounding an organism.

Moving on to the projection matrix of the VISpm area, it can be observed that the projection density of the VISpm region towards the RSP region is much greater than the projection density of the VISpm to any other region, again with the only exception being the towards the VIS region which is addressed above. Our findings that the VISpm region has a high projection density towards the RSP region is supported by a 2018 study in which Harris demonstrated that the projection density between the RSP region and the VISp region is much higher than the projection density between the RSP region and the VISal region (Harris, 2018). The data was depicted on a logarithmic heat map, therefore the projection data is without units. Although there is no clear consensus on the precise function of the RSP region, commonly referred to as the retrosplenial cortex, several studies both on human and animals point to a role in learning landmark locations and the consolidation/retrieval of spatial information (Mitchell et. al, 2018). A possible reasoning supported by the literature as to why we observed a very high projection density between the VISpm region and the RSP region is that the neurons of the VISpm region takes visual stimuli to the RSP in order to retrieve spatial information given the context of their surroundings in order to locate where they are, or to encode spatial information into the RSP region in order that they may retrieve that information later.

Finally, DAVID analysis was used to in order to analyze the difference in gene expression, the primary difference that was found according to the enrichment scores was that the anterolateral area has very high gene expression in genes that code for neurotransmitter receptors (such as serotonin, oxytocin, dopamine, and several others) when compared to posteromedial area. On the other hand the posteromedial area had very high enrichment scores and gene expression in genes that coded for the Golgi apparatus and the Golgi membrane, which is responsible for the packaging of proteins, and very high gene expression in the cAMP pathway. This points to the possibility that the neurons of the posteromedial area of the brain have a G-protein signaling mechanism/pathway that is integral to their function. The literature was extensively searched to find an explanation for

this, however to the best of our knowledge no research has yet been done on the signaling pathway of the VISpm. This would be an excellent candidate for future research subjects.

6.6 Limitations

In our experiment we only focused on mice. Although mice are good model organisms for the visual system, mice lack the cortical regions compared to humans and other primates. As a result, the result can not extrapolate visual processing for humans . For the violin plot (Figure 2), the differences in shape arises from the number of neurons; AL had less cells compared to PL . This finding will lead to the difference in shape. The data that was obtained for the projection matrices of the two regions are difficult to quantify in a precise manner as they are depicted in a unitless logarithmic scale. Finally, there is a small sample size of neurons from the anterolateral and posteromedial region and if we continue to segregate these two regions through cre lines there is a very limited number of cells with each cre line, making comparison between the electrophysiology features of the cells unreliable because there are too few cells to make the results statistically insignificant.

6.7 Future Direction

We may look more into the connectivity of higher visual areas to see if there are inputs into the VISal area and the VISpm area. We may also do RNA sequencing on the VISal and VISpm areas to further quantify the highly expressed clusters. Finally, we may record temporal electrophysiology features for the VISal and VISpm cells so that we may have more precise clustering data for slow moving and fast moving stimuli.

6.8 Reflection

NAME:

This has been a really fun project for me. I did not expect that we can do a proper research on specific higher visual areas in the visual cortex. This is exciting to mine from various datasets that contain different features of the same types of cells: ranging from genetic information, electrophysiology features, 2-photon imaging to connectivity. Even thought we did not get significant distinguishable results from the two regions, I see that this is a viable model for studying different regions across the brain

Reflecting from Dr. Voytek's talk, I started to understand more about the definition of data science: the ability to clean and extract the data in datasets, and connecting them with a common language.

The plan of the project is integrating the existing datasets on Allen database, and one of the datasets I anticipate the most in the genetics dataset, since I hope to get a higher resolution of genetic information from the RNA-seq technique. However, we did not find good categorizations of brain areas, and we could only reside on ISH data.

Moreover, I think the most difficult part of the project is to pool out the right cells. The difficulty lies in finding the right balance: if it's too specific, there might only be 1/2 cells available in the dataset; if it's too broad, then the information might be too varied to gain insights.

All in all, I really enjoy the procedure of finding a research question, exploring the relevant datasets and actually visualising the data. I could expect that with the valuable skills I learned in this class, I will be shifting towards a more data-oriented approach when tackling neuroscience questions.

NAME:

This project was interesting in that I was able to look for other research papers that had findings that were either similar or dissimilar to our findings. I found the project to be quite difficult as someone who is completely new to coding, however looking through the code and learning how we can apply what we have learned in class to produce meaningful data is quite rewarding. The most difficult part of the project would be where we had to go back and iron out a lot of issues that arose in our code or in our analysis over and over again. The literature on these two areas is quite scarce as well so determining the functional specialization of the two regions was tricky. The most rewarding part of this project was putting forth my best effort in tackling the challenges that came with this project, especially digging into the literature to find justifications for our results and tying together several of our results to find a justification for our results.

NAME:

The project was a lot of work and took up more than 15 hours outside of class but I learned a lot from this project . The project required me to meet with the group multiple times a week. We ran into a lot of problems pulling data and visualizing the data. Being new to coding, I heavily relied on NAME on helping me do the data wrangling and data visualization. Another problem is coordinating putting the journal together. We had to message each other back and forth to coordinate changes to the journal. Overall, the project was interesting. I learned a lot from it.

[ ]: