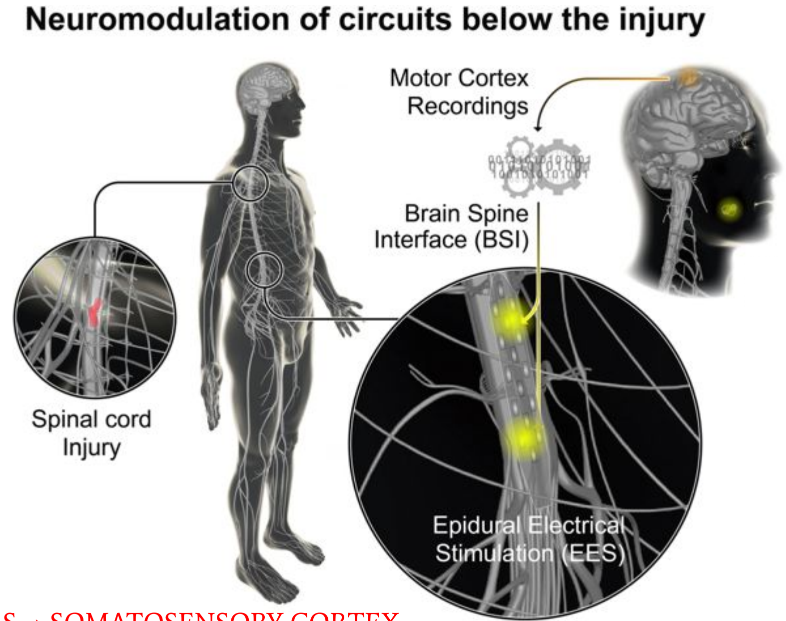
Classifying Evoked Potentials from Spinal Cord Microelectrode Array Data

Jeremy Ford, Samantha Russman, Margot Wagner
COGS 260
Spring 2020

Brain-spine interfaces for treatment of spinal cord injury

A wireless **brain-spine interface (BSI)** could restore movement and sensation in patients with spinal cord injury.

Spinal cord electrical stimulation and recording **via microelectrode arrays** can relay information between the brain and the body.



Cho et al. 2019

Two pathways must be restored:

1. Sensory information

SENSORY NEURONS (LIMBS) → SPINAL CORD SENSORY EVOKED POTENTIALS→ SOMATOSENSORY CORTEX

2. Motor information

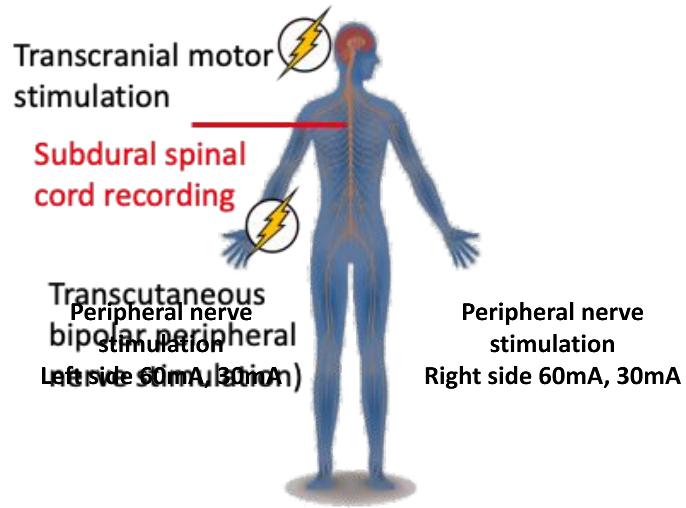
MOTOR CORTEX → SPINAL CORD MOTOR EVOKED POTENTIALS → MOTOR NEURONS (LIMBS)

Transmitting sensory information to/from the spinal cord requires classification of sensory evoked potentials.

Experimental setup & collected data

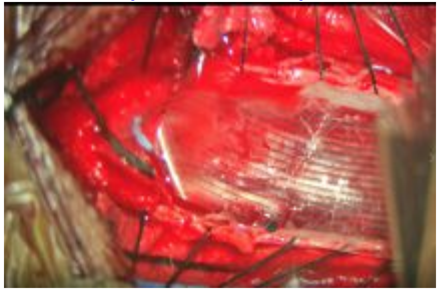
We performed peripheral sensory nerve stimulation in a patient undergoing a spinal cord tumor resection and recorded from the spinal cord with a microelectrode array with 100s of contacts.

Stimulation varied based on side of the body and stimulus level and was administered with neurophysiology monitoring equipment.

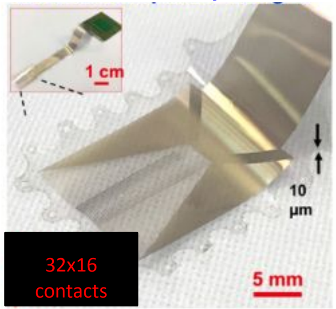


Can a supervised learning model classify different types of sensory evoked potentials from recordings of electrical activity in the spinal cord?

Electrode placement on spinal cord

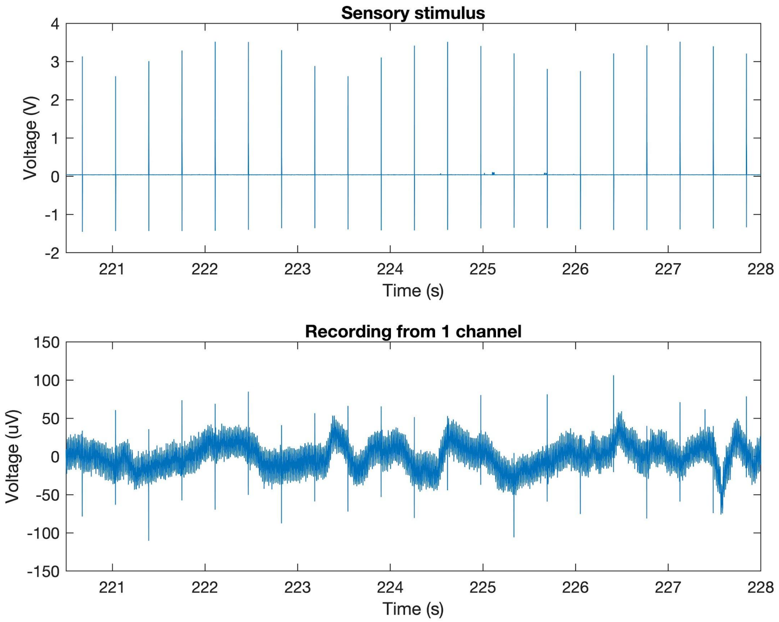


Electrode layout

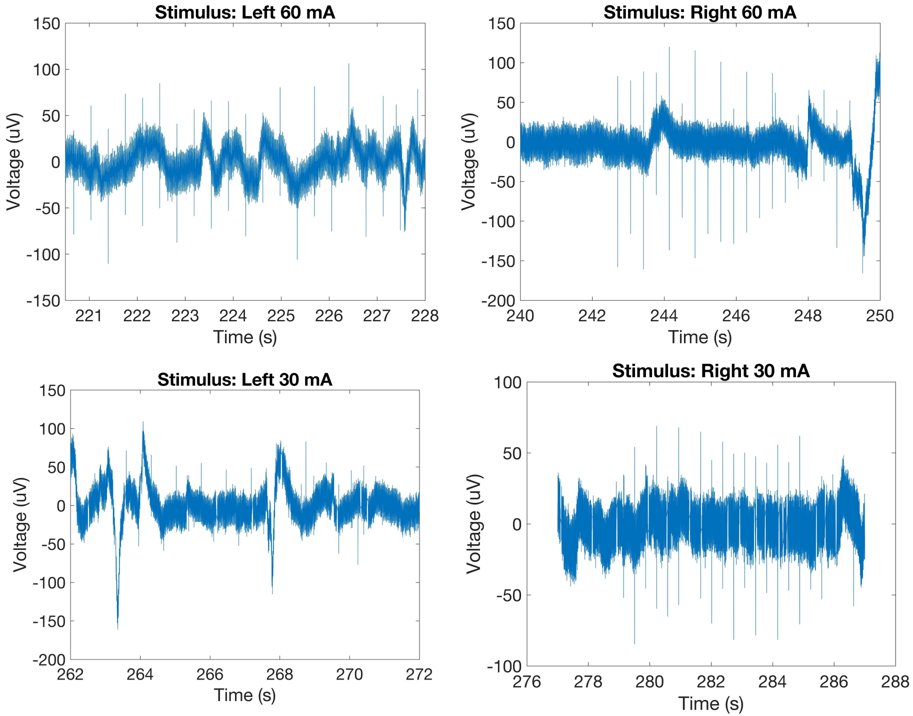


Data preprocessing

1. Alignment



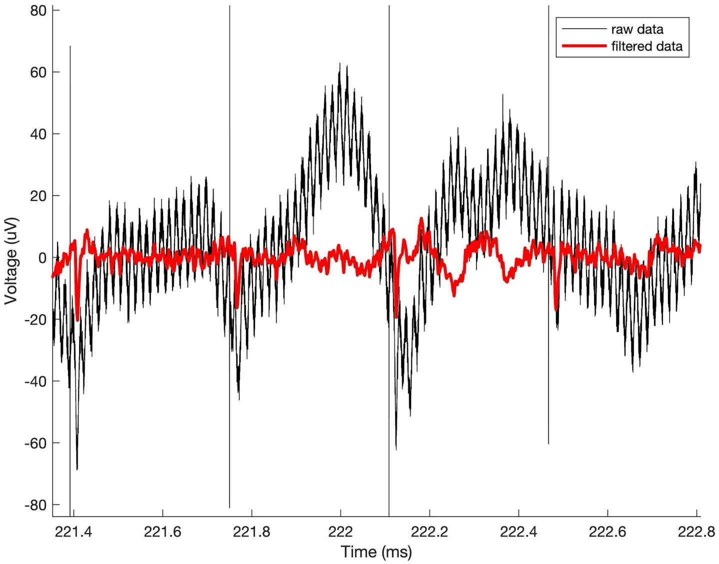
2. Windowing



Data preprocessing

3. Filtering

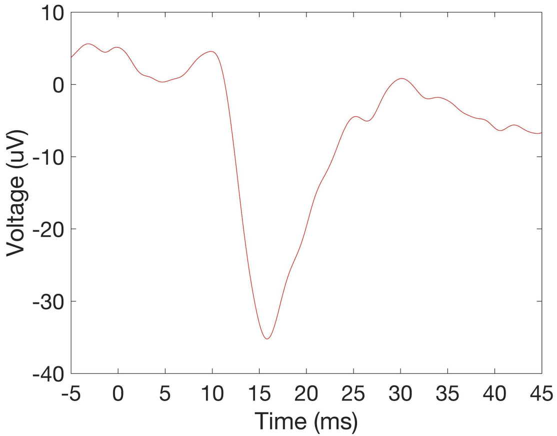
- Notch filters for 60 Hz noise and harmonics
- Bandpass filter [3 300 Hz] → captures LFP behavior



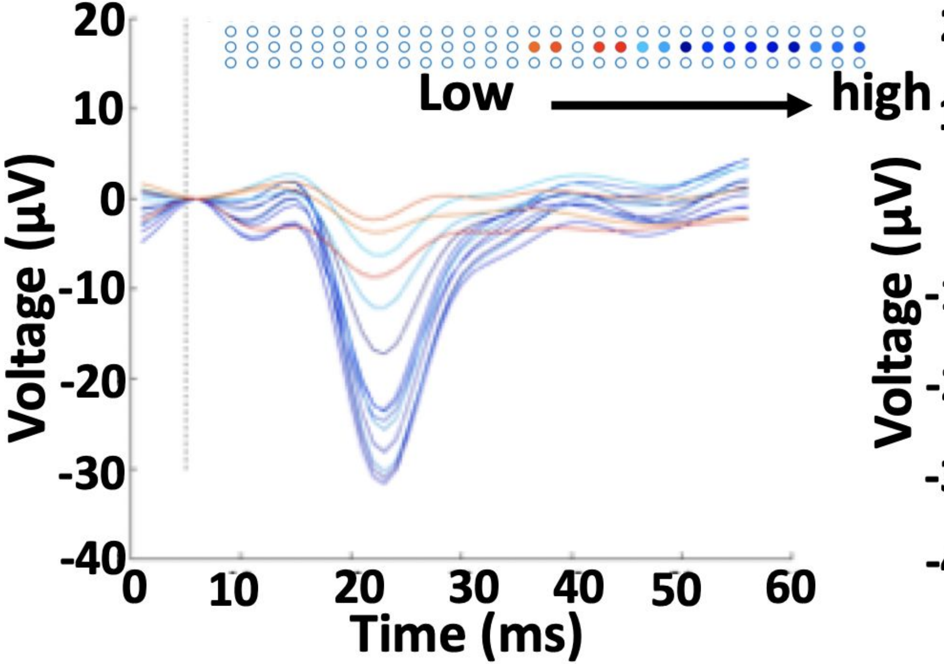
4. Initial bad channel selection

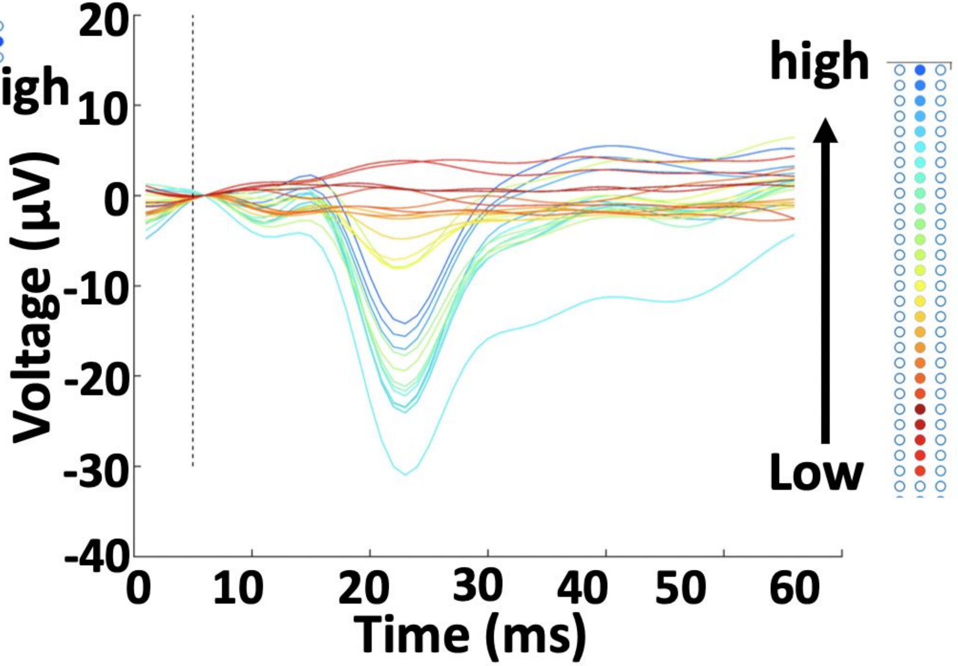
- Based on impedance measurements for all channels.
- Impedances below 150 kOhms
- Resulted in 353 “good channels”

5. Segmentation into individual trials

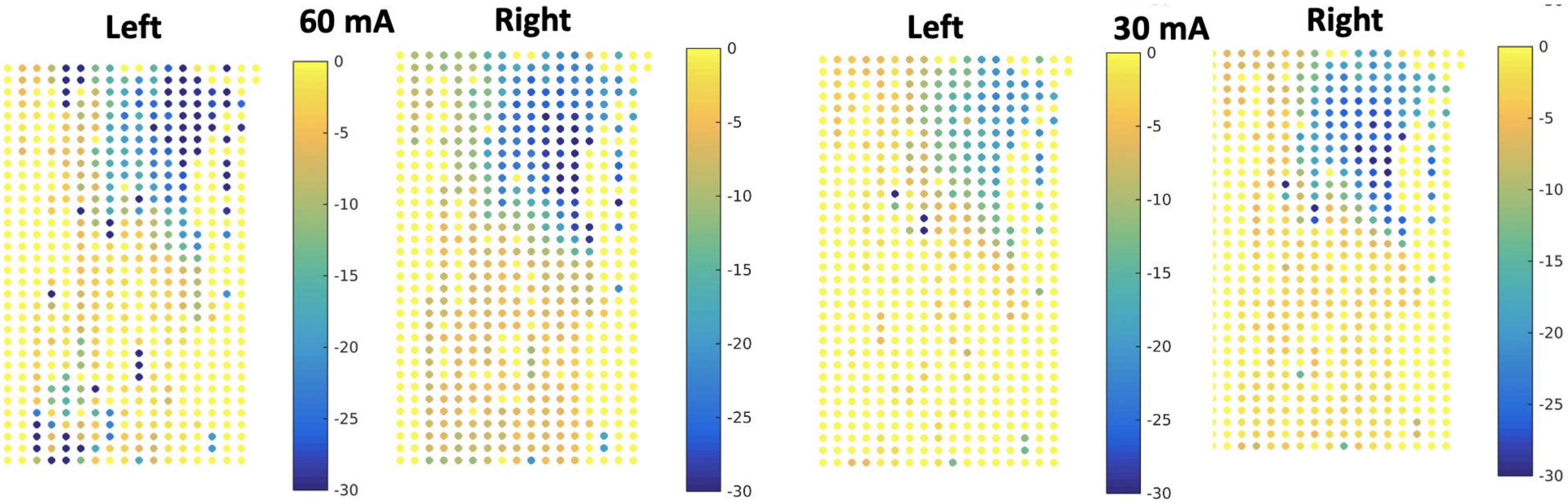


Assessing recordings before classification





Assessing recordings before classification



Hypothesis & project goal

Goal: Detect and classify single evoked potentials in a short time-window of data recorded from a 300+ channel microelectrode array placed subdurally on the spinal cord surface.

- The five classes of interest are:
  - sensory evoked potentials at 60 mA stimulation (left and right)
  - sensory evoked potentials at 30 mA stimulation (left and right)
  - no evoked potentials (baseline)

- Input dataset is 50 ms long and sampled at 20 kHz

- Initial hypothesis: extracting the top principal component (PC) from the input data will be sufficient to achieve a 50% classification accuracy rate via a multiclass linear discriminant analysis.

Analysis - Outline

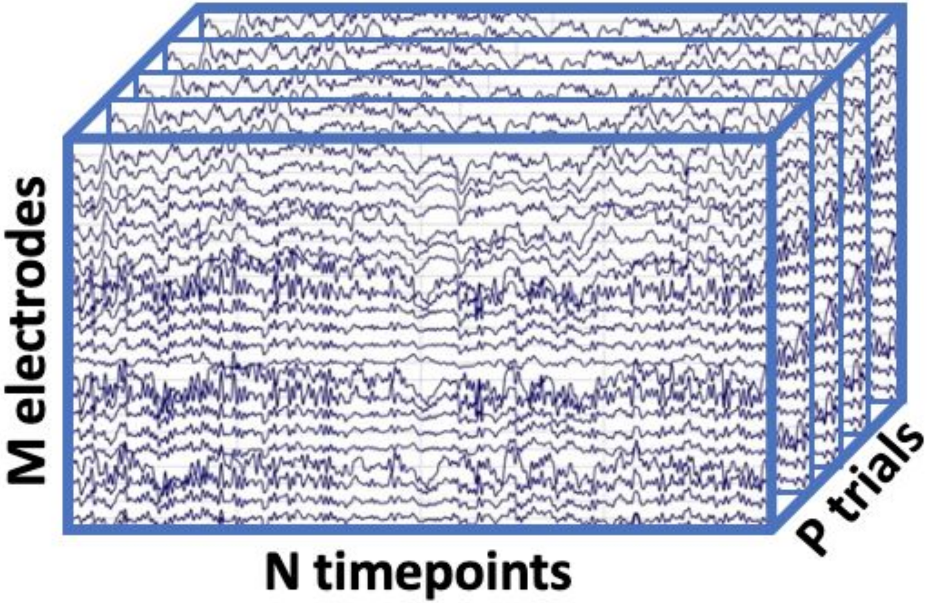
1. Preprocessing
2. Dimensionality Reduction and Feature Extraction
3. Classification

Analysis - Outline

1. Preprocessing
2. Dimensionality Reduction and Feature Extraction
3. Classification

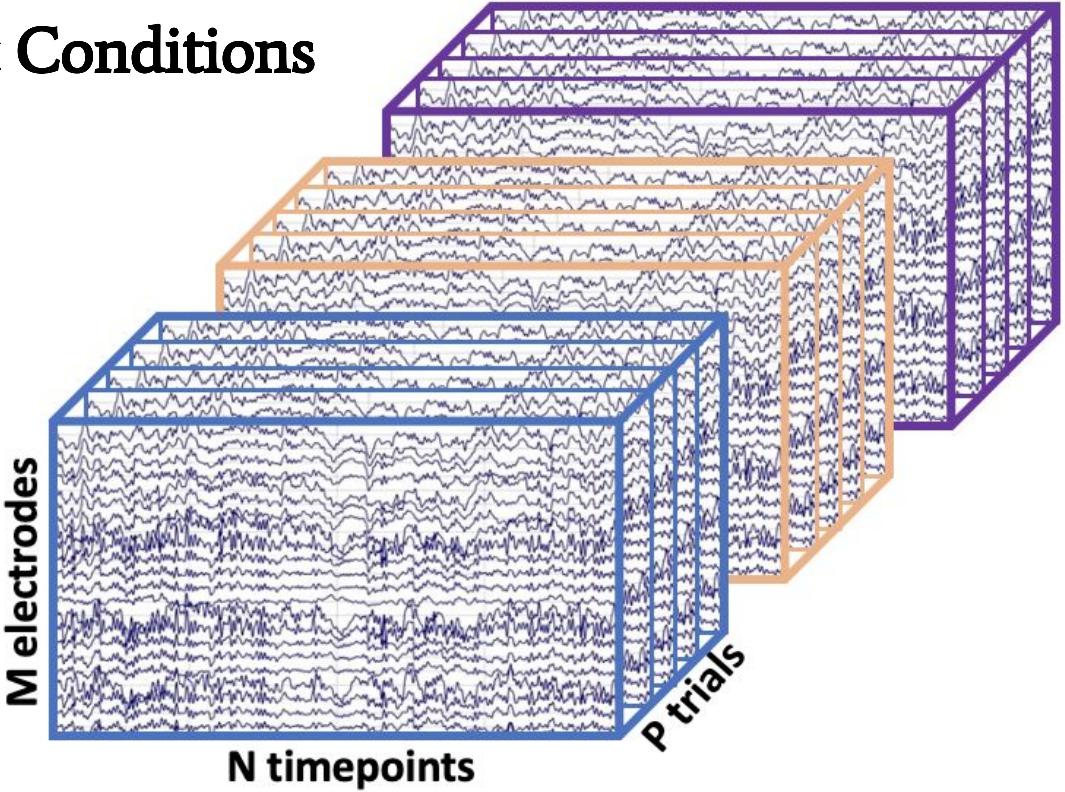
Analysis - Data Matrix

1 Condition



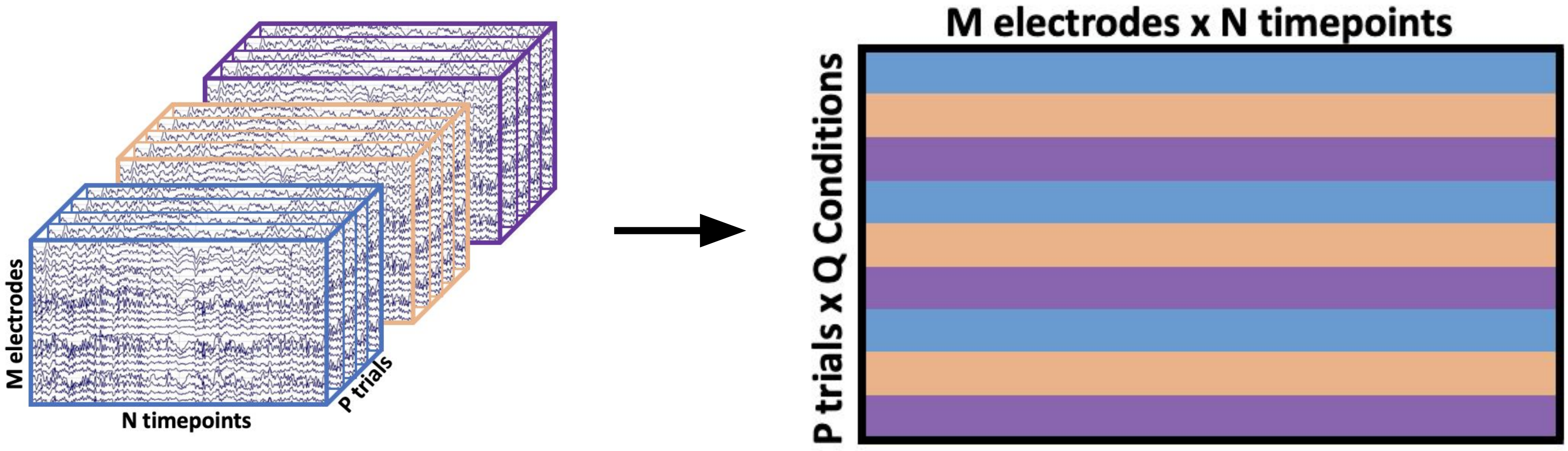
Analysis - Data Matrix

D Different Conditions



Analysis - Data Matrix Reshaping

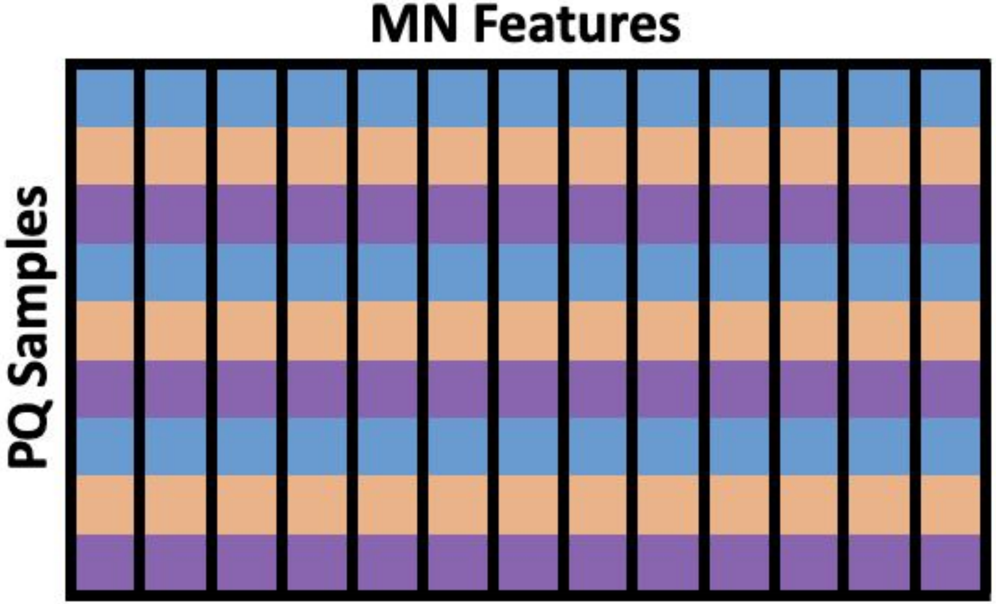
Dimensionality reduction requires a 2D matrix



Analysis - Data Matrix Reshaping

Samples x Features

- *Mean centered data:* zero mean across features



Analysis - Outline

1. Preprocessing
2. Dimensionality Reduction and Feature Extraction
3. Classification

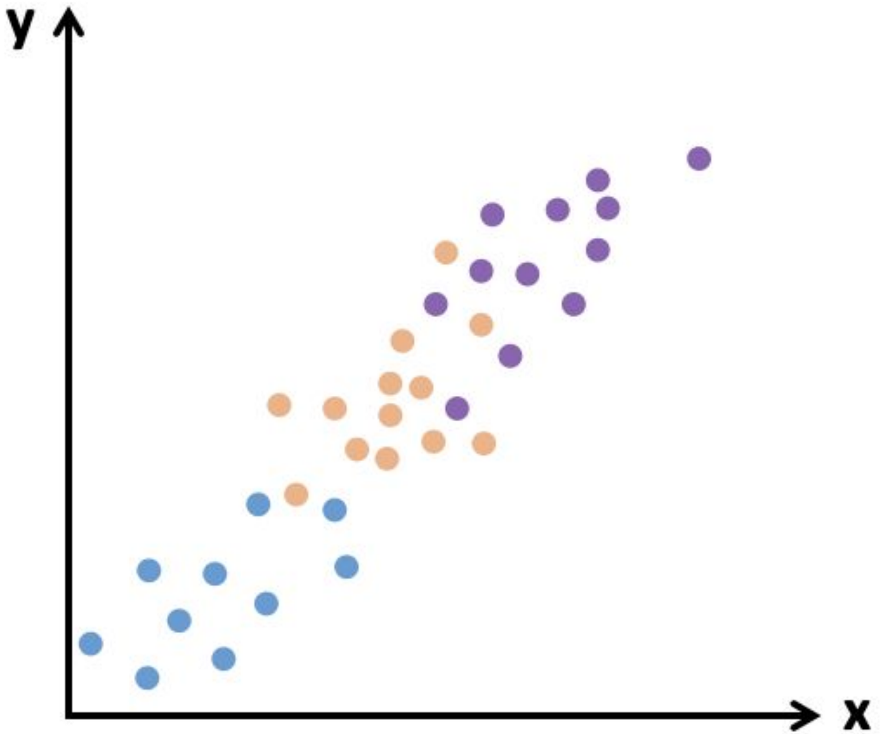
Analysis - Principal Component Analysis (PCA)

- **PCA:** Dimensionality reduction technique decomposing the data into variance directions and magnitudes and keeping the directions of highest variance.
- Keeps the most "informative" dimensions
- Unsupervised - no labels

Analysis - Principal Component Analysis (PCA)

Dimensionality reduction

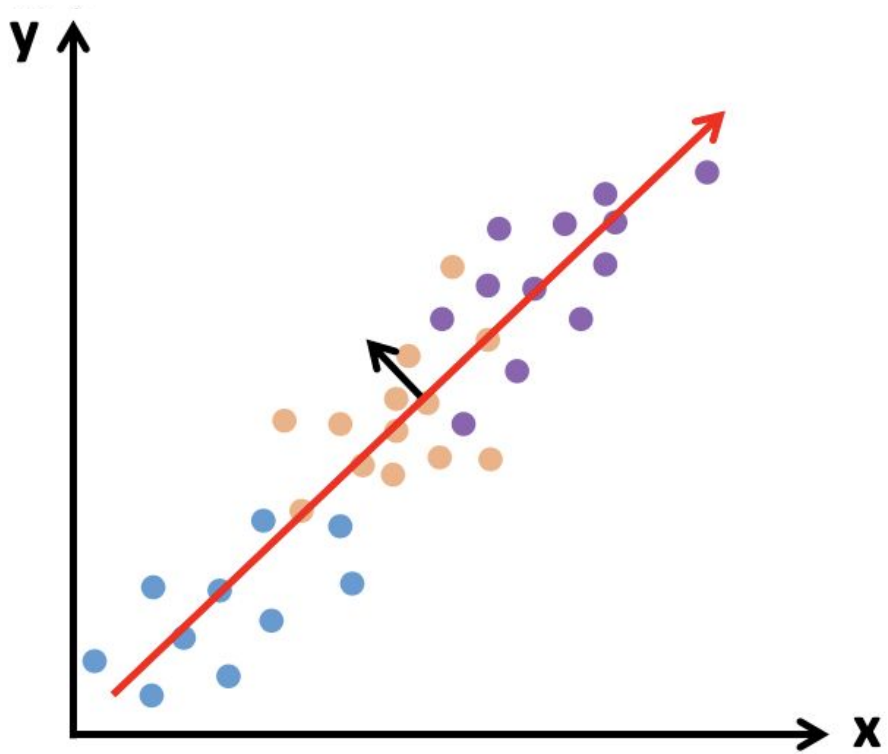
1. Create new uncorrelated variables (principal components) that maximize variance
    - These variables are linear functions of the original variables



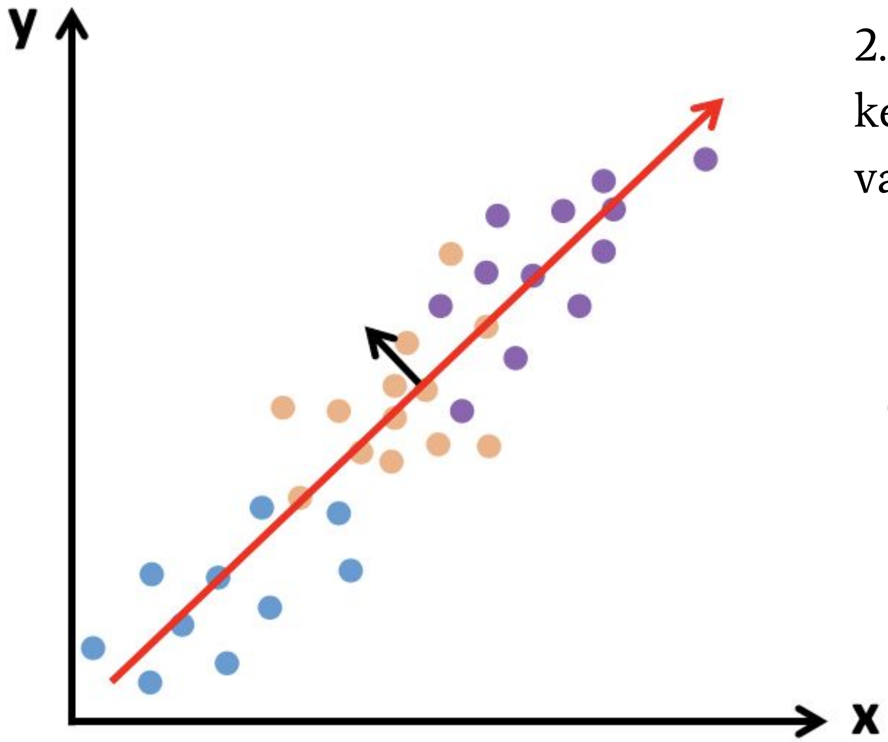
Analysis - Principal Component Analysis (PCA)

Dimensionality reduction

1. Create new uncorrelated variables (principal components) that maximize variance
   - These variables are linear functions of the original variables

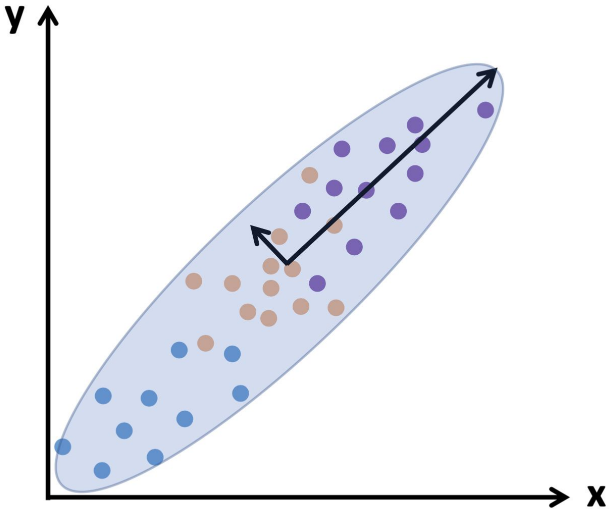


Analysis - Principal Component Analysis (PCA)



2. Reduce the dimensionality of the dataset by keeping the components with the most variance, or spread.

Analysis - Singular Value Decomposition (SVD)



$$X = U\,S\,V^T$$

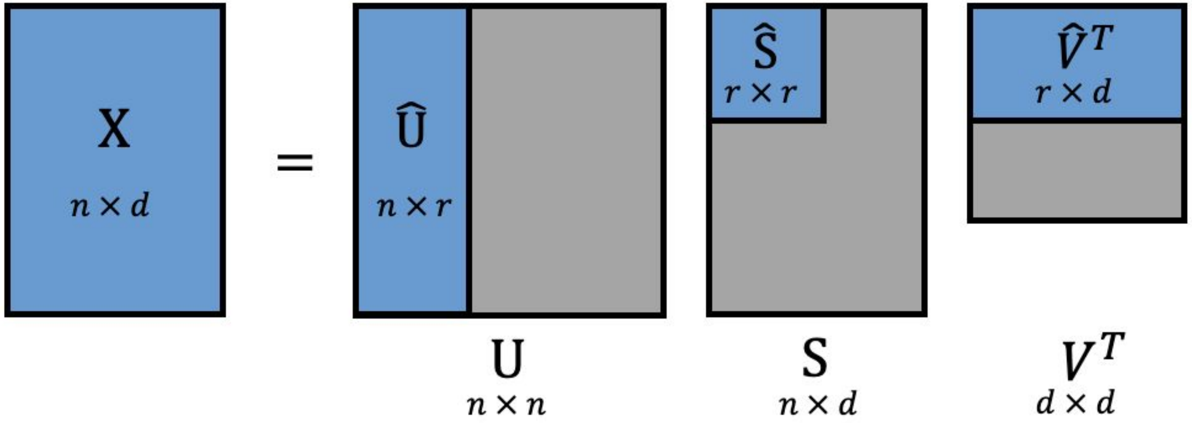
https://en.wikipedia.org/wiki/Singular_value_decomposition

Analysis - Singular Value Decomposition (SVD)

- For an *n x d* data matrix **X**:
  - **U:** Eigenvectors of XX^T
  - **V**: Eigenvectors of X^TX - Principal directions or axes
  - **S**: Singular values - Eigenvalues of XX^T - diagonal matrix
- Can limit the resulting size of the matrix - $r \leq \min(n, d)$
- **Transformed data** (principal components): XV = US

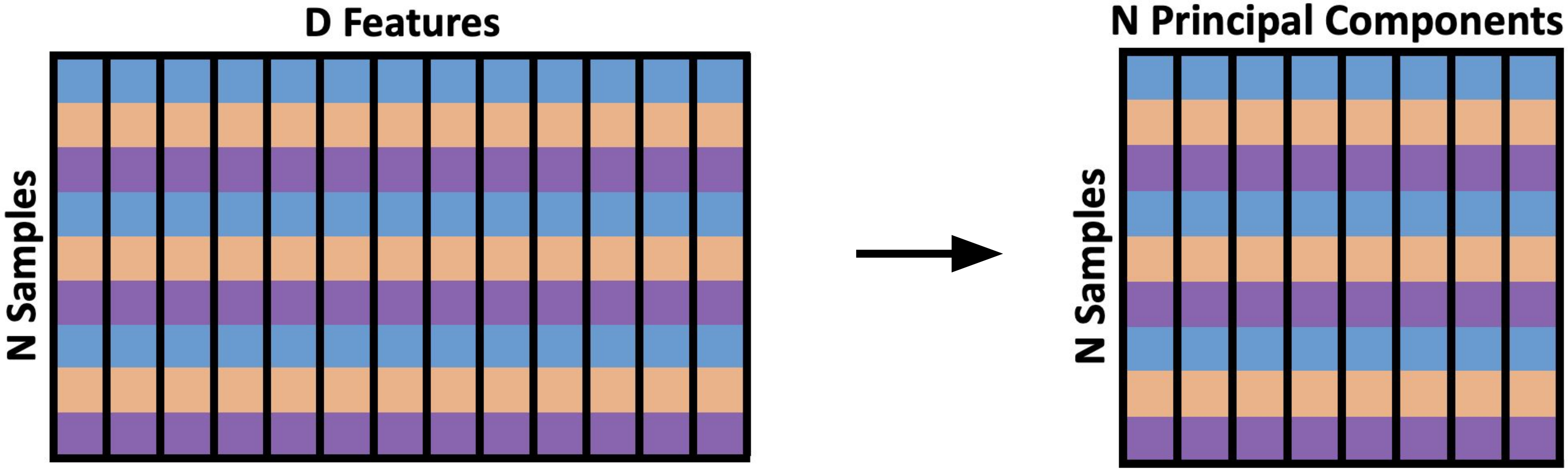
$$\mathbf{X = U\,S\,V^T}$$

$$XV = US$$



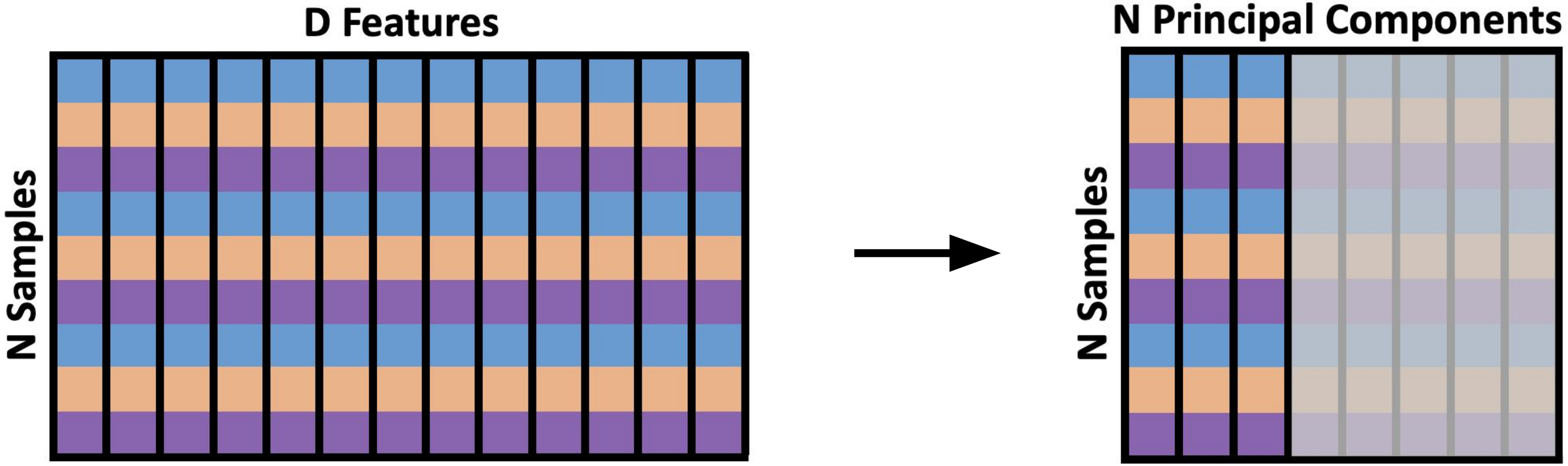
Analysis - Principal Component Analysis (PCA)

- Transform data onto principal axes
- Select a subset of principal components to reduce dimensionality while being able to explain data

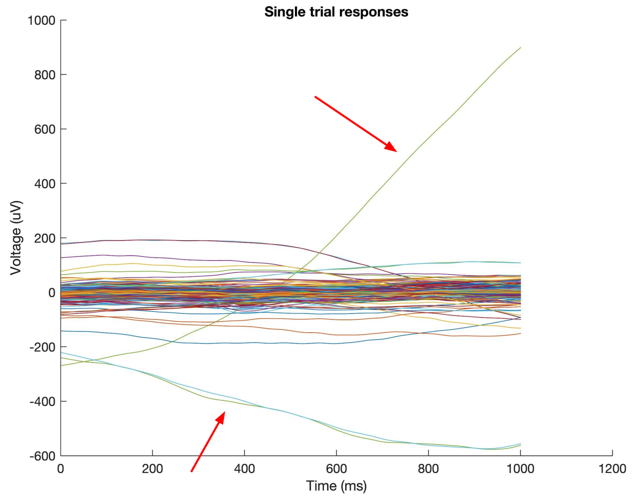


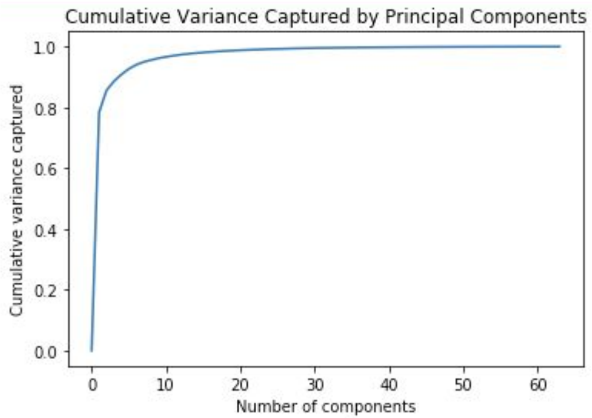
Analysis - Principal Component Analysis (PCA)

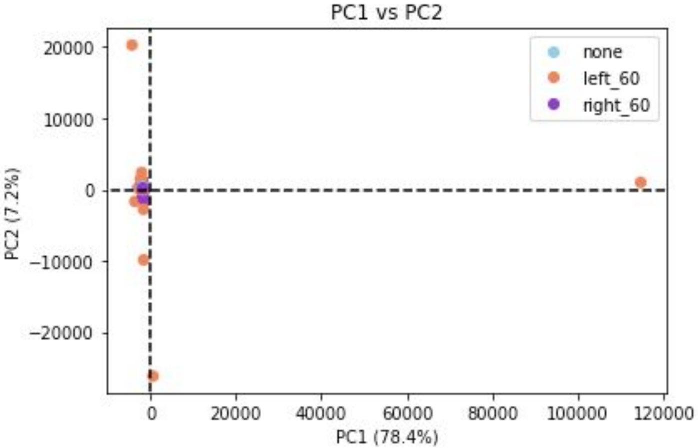
- Transform data onto principal axes
- Select a subset of principal components to reduce dimensionality while being able to explain data

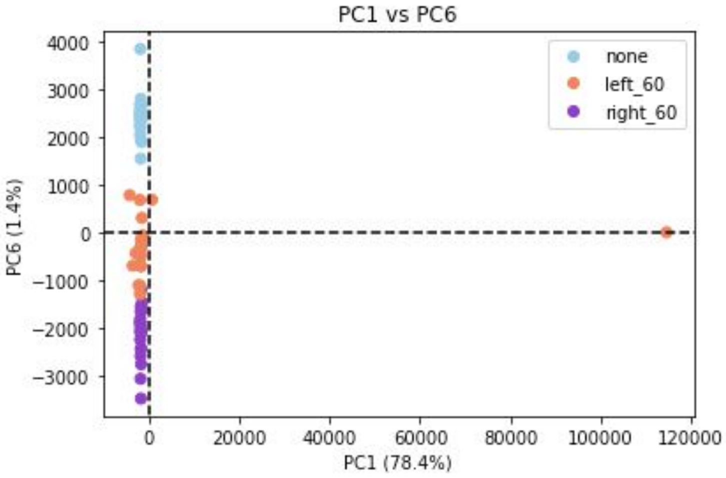


Analysis - PCA Outlier Detection

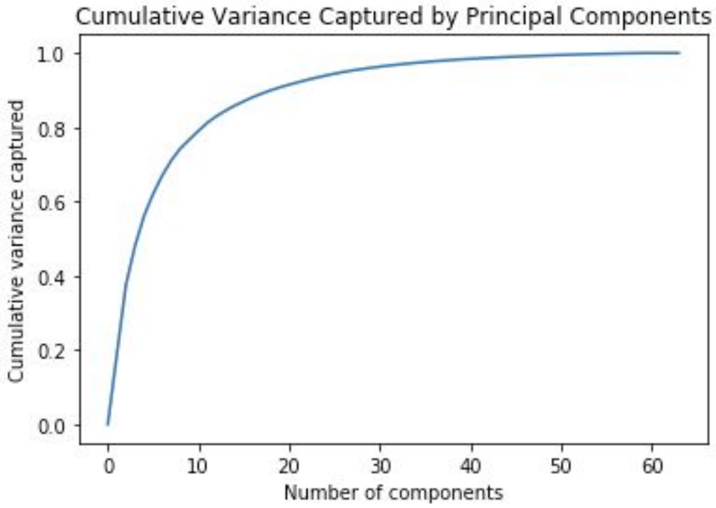


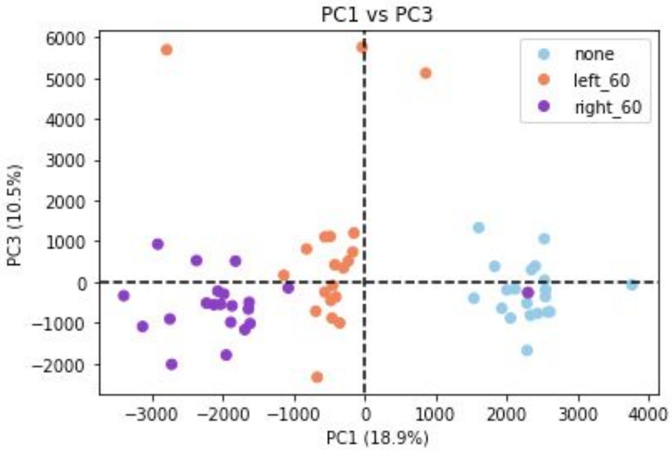


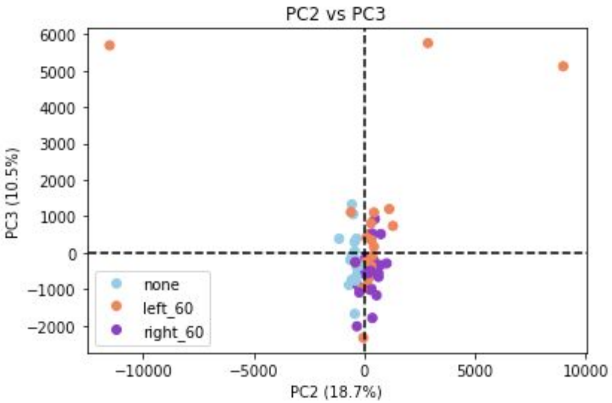


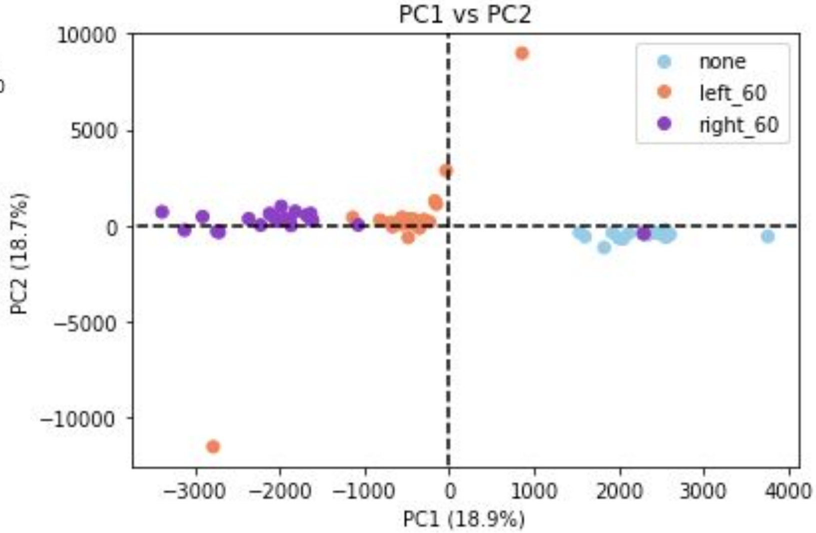


Analysis - PCA 3 Conditions

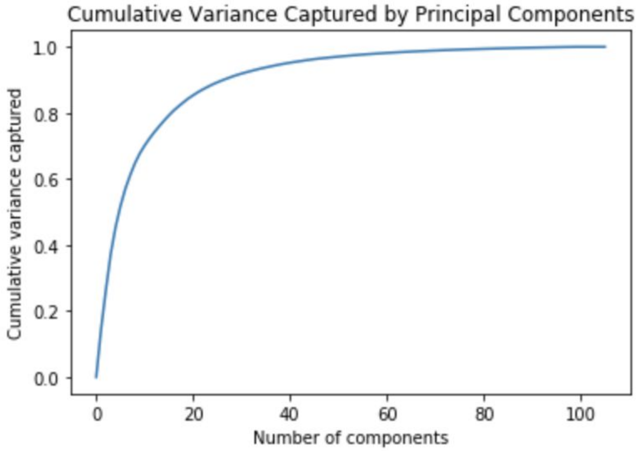


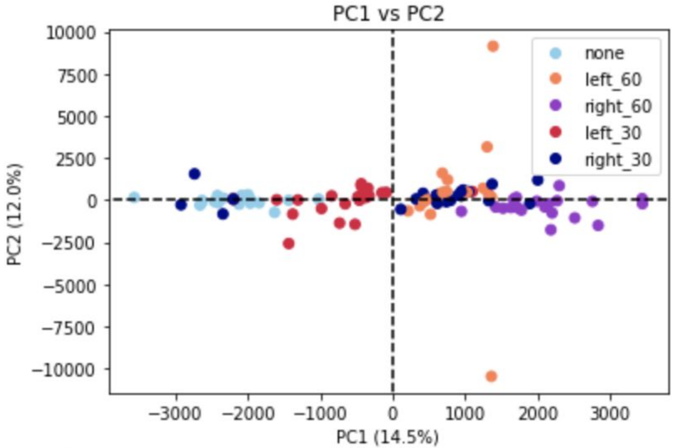


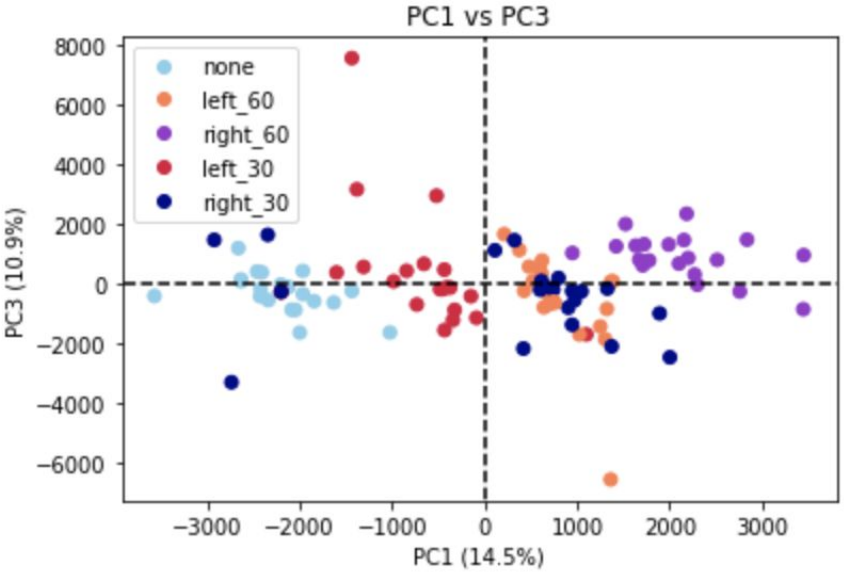


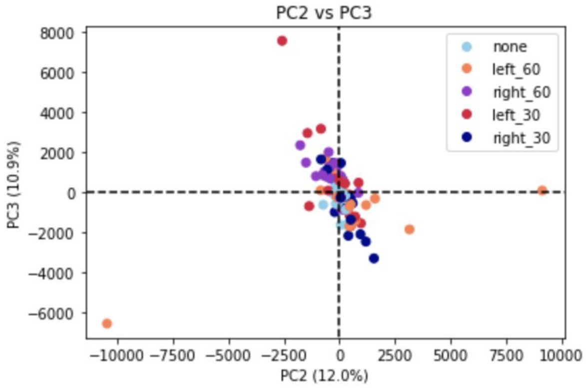


Analysis - PCA 5 Conditions









Analysis - Outline

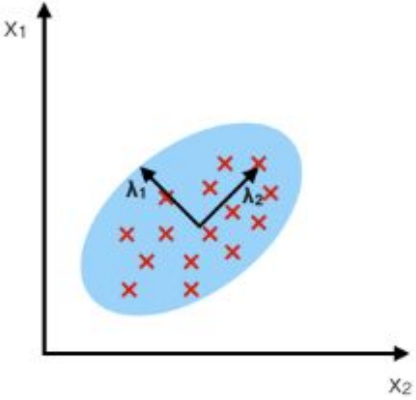
1. Preprocessing
2. Dimensionality Reduction and Feature Extraction
3. Classification

Analysis - Linear Discriminant Analysis (LDA)

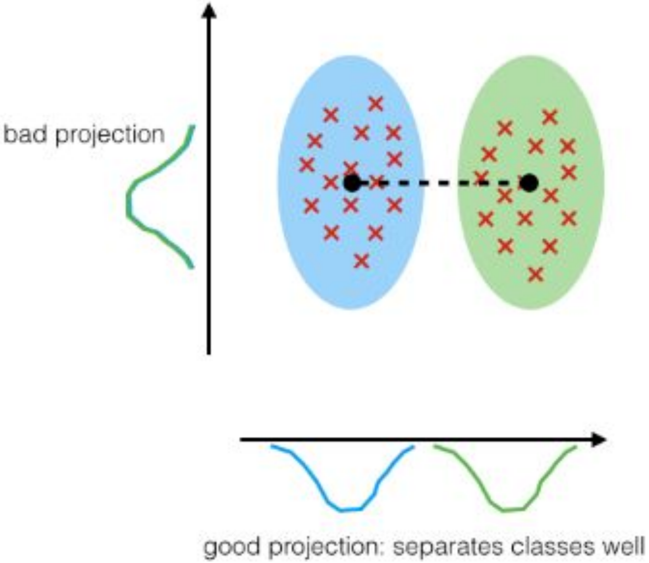
- Dimensionality reduction technique
- Determine axes that maximize the separation between classes
- Supervised - classes are labeled
- Can be combined with PCA
- Useful for classification

Analysis - LDA

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation

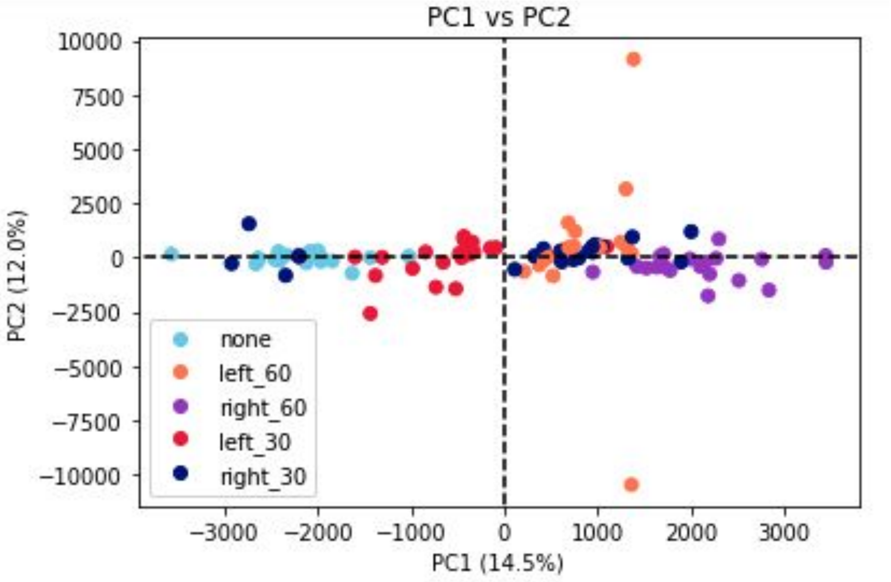


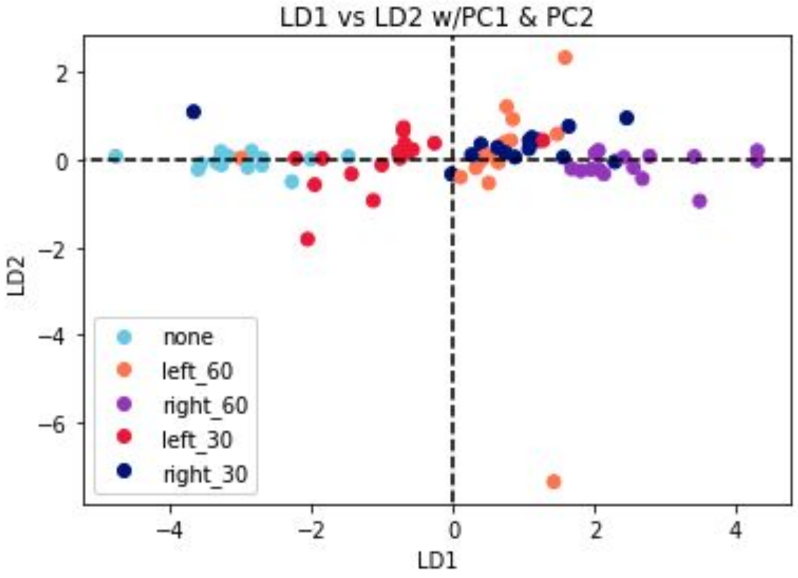
http://sebastianraschka.com/Articles/2014_python_lda.html

Analysis - LDA Implementation

1. Compute the mean vectors for each class
2. Compute the scatter matrices
   2.1. S_w - within-class scatter
   2.2. S_b - between-class scatter
3. Solve eigenvalue problem for $A = S_w^{-1}S_b$
   3.1. $Av=\lambda v$
4. Select linear discriminants
   4.1. Sort eigenvectors by decreasing eigenvalues
   4.2. Keep top-n eigenvectors
5. Transform input to the LD space

Analysis - PCA vs LDA





Analysis - Naive Bayes Classifier

- Bayes Theorem
- Estimate probabilities from LDA
- Take class with highest conditional probability

$$P(\mathcal{C}_j|x) = \frac{P(x|\mathcal{C}_j)P(\mathcal{C}_j)}{P(x)}$$

$$\text{Decide } \begin{cases} \mathcal{C}_1 & \text{if} \quad P(\mathcal{C}_1|x) > P(\mathcal{C}_2|x) \\ \mathcal{C}_2 & otherwise \end{cases}$$

https://web.fe.up.pt/~jsc/ML/lectures/lecture03.pdf

Analysis - K-Fold Cross-Validation

- Split data into k sets
- Iterate through test/train k times, swapping which set is used for testing each iteration
- Helps to reduce bias in the estimate of the model's performance

https://commons.wikimedia.org/wiki/File:K-fold_cross_validation_EN.svg

Analysis - Classifier Metrics

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

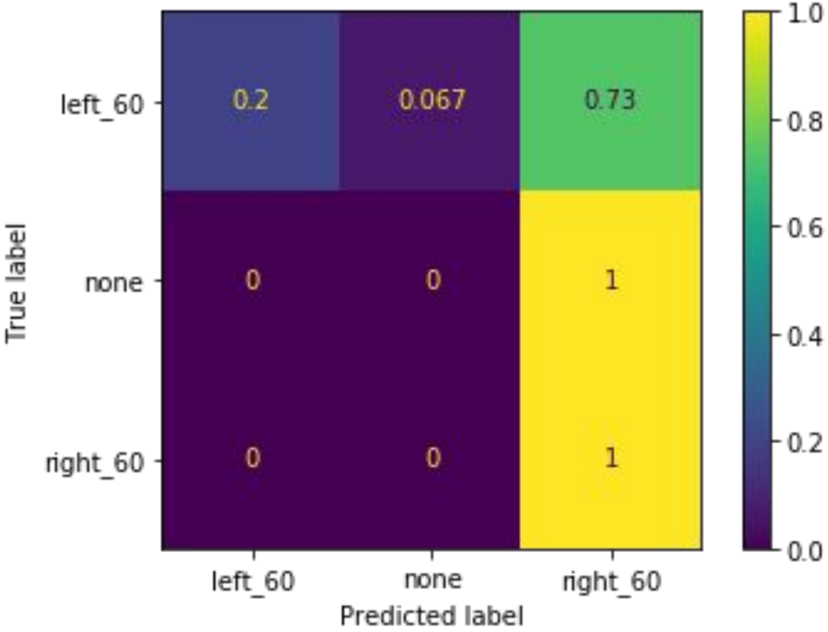
$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

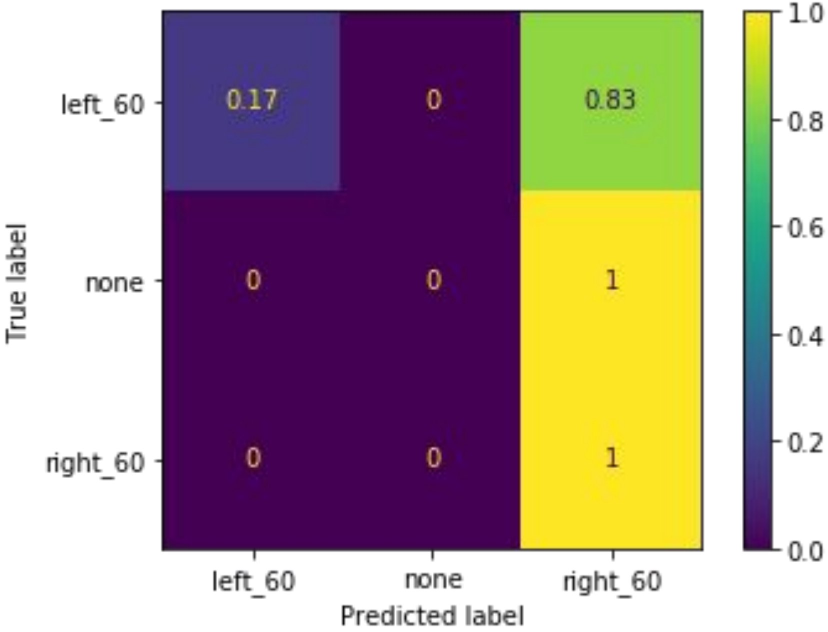
$$F1\ score = \frac{2 * precision * recall}{precision + recall}$$

Results - 353 Channels, 3 Classes

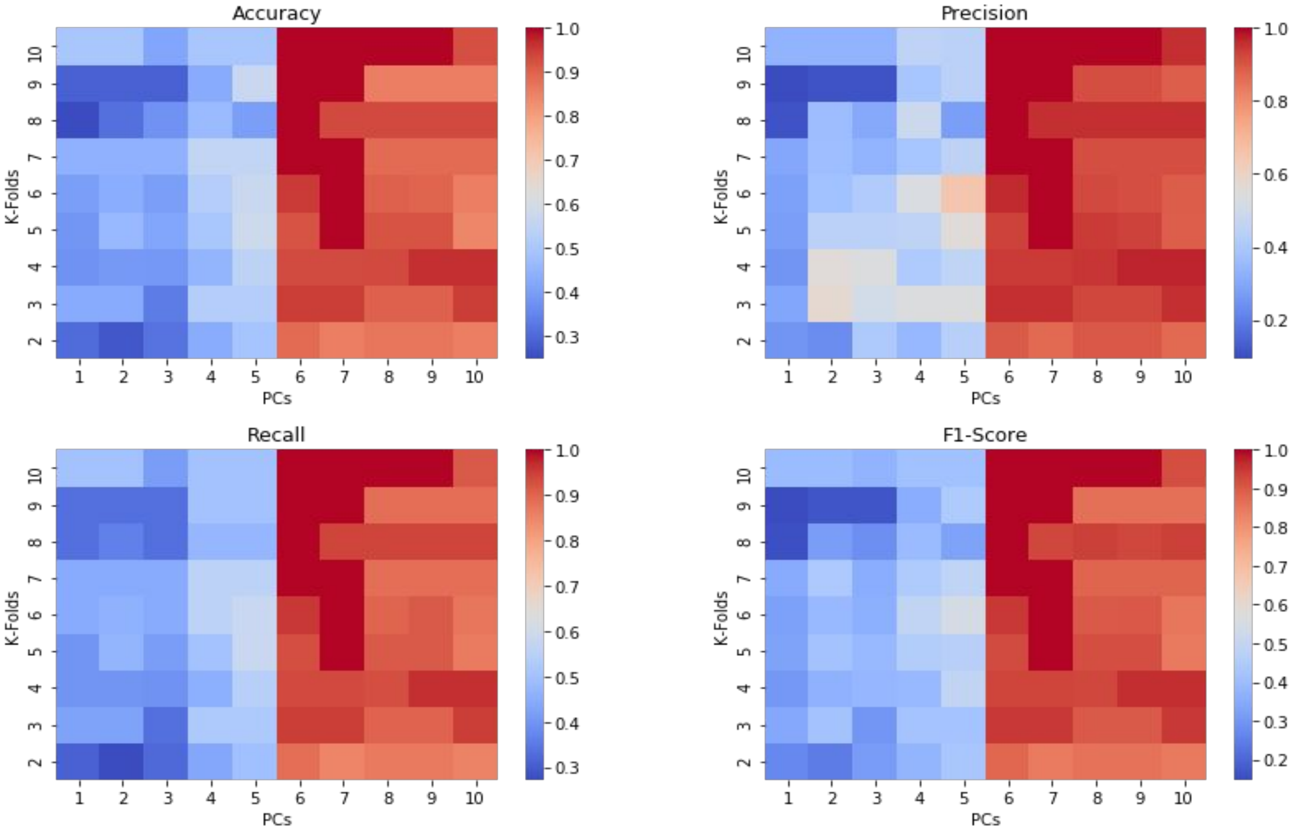
Training



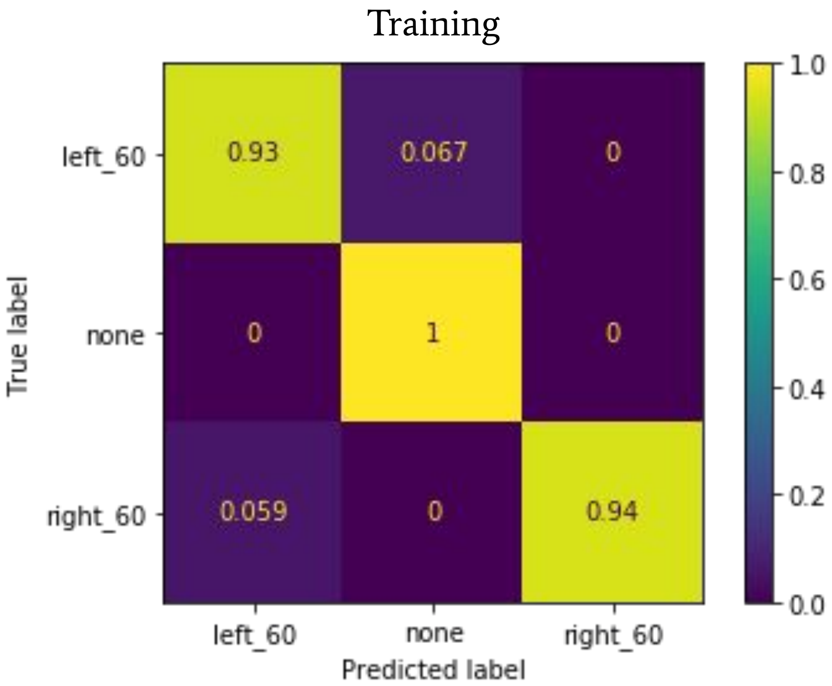
Testing

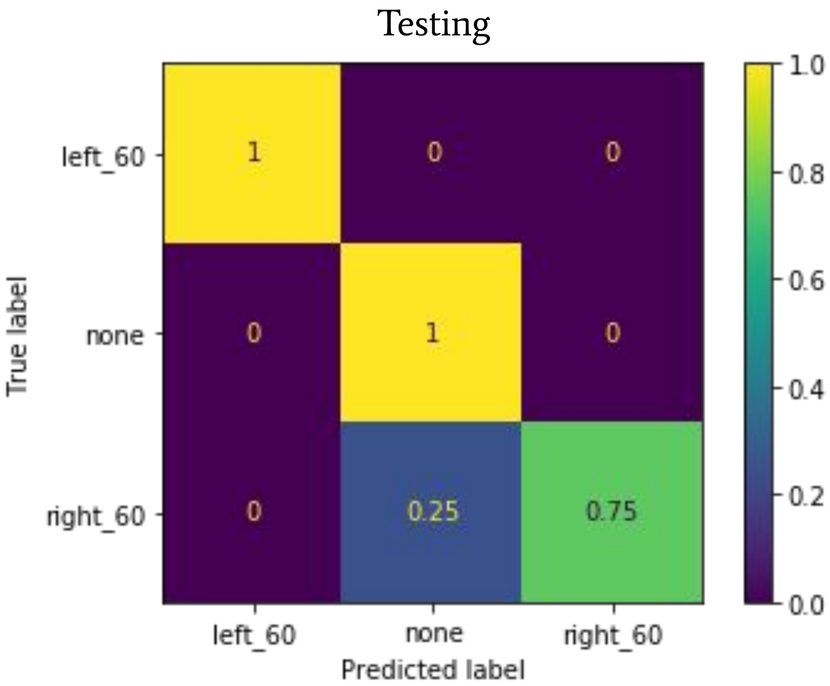


Results - 353 Channels, 3 Classes

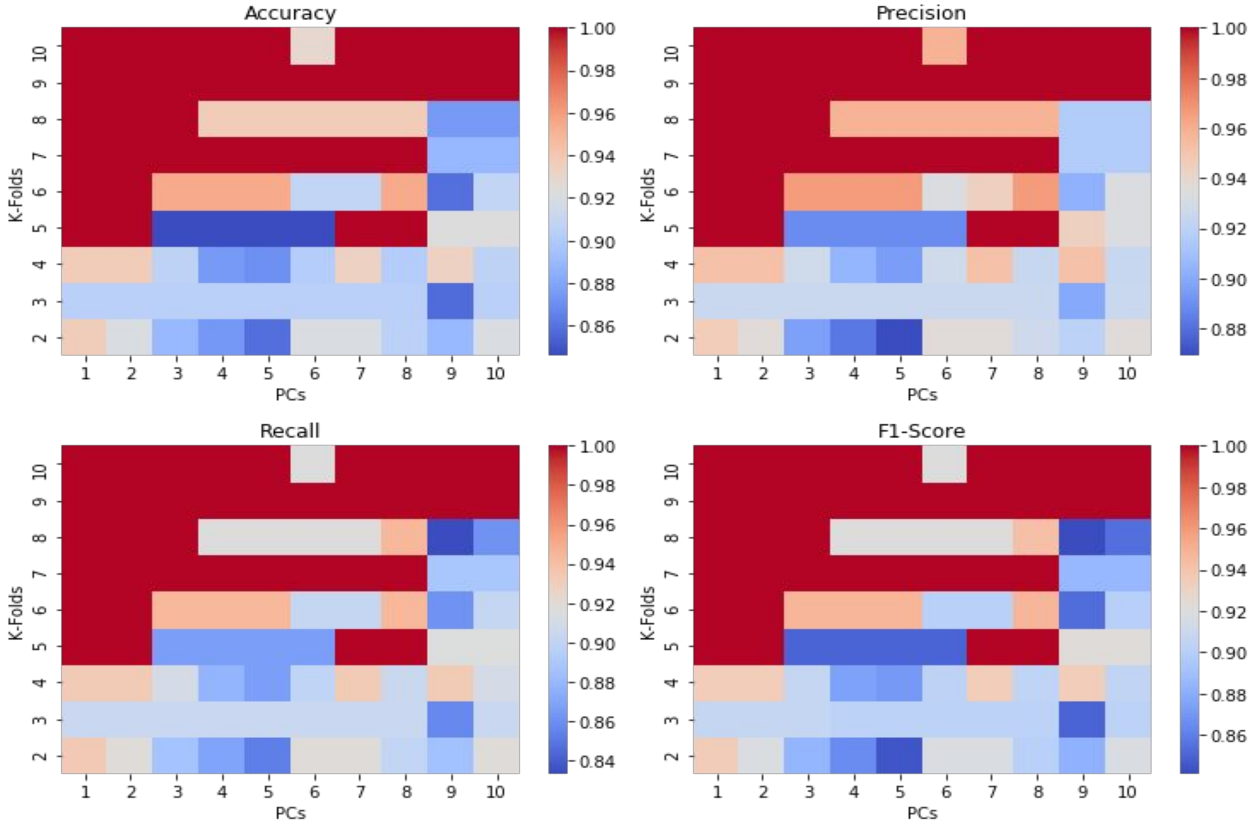


Results - 310 Channels, 3 Classes



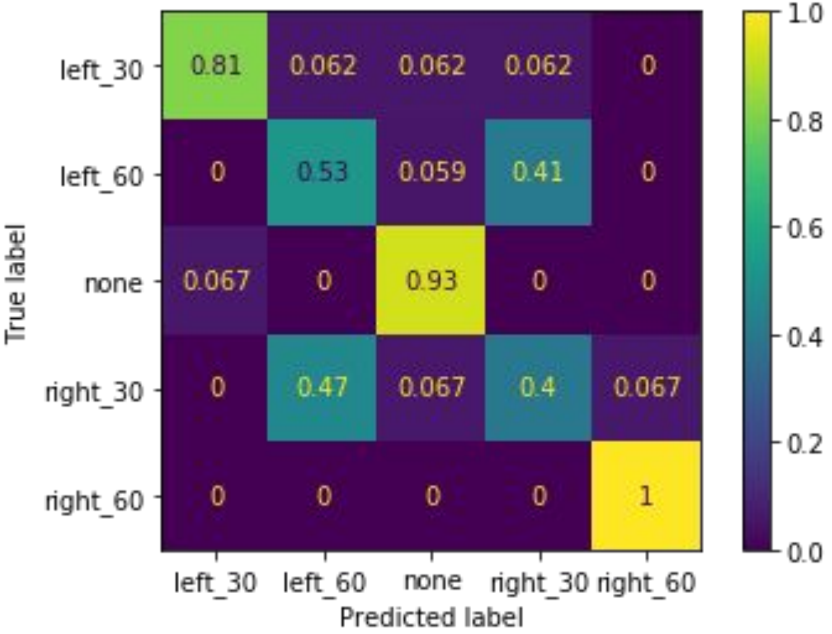


Results - 310 Channels, 3 Classes

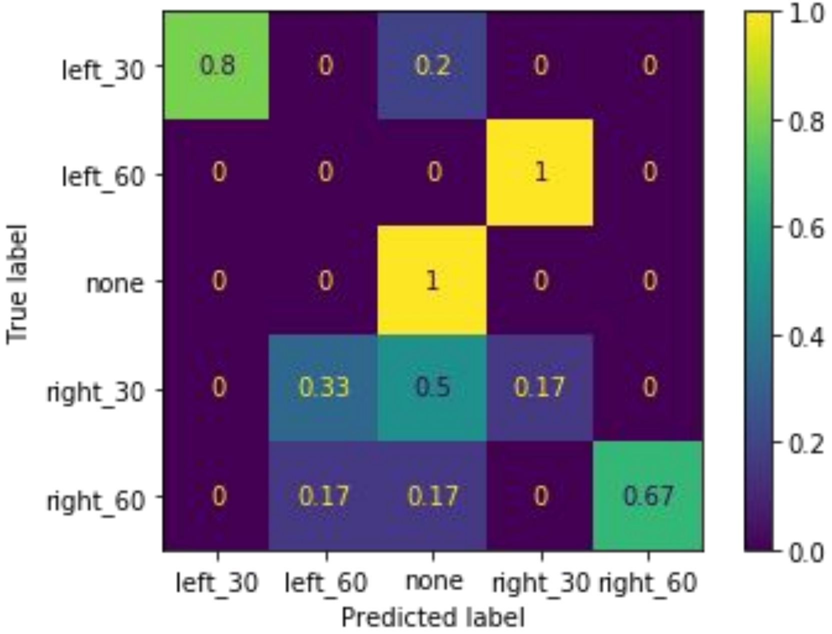


Results - 310 Channels, 5 Classes

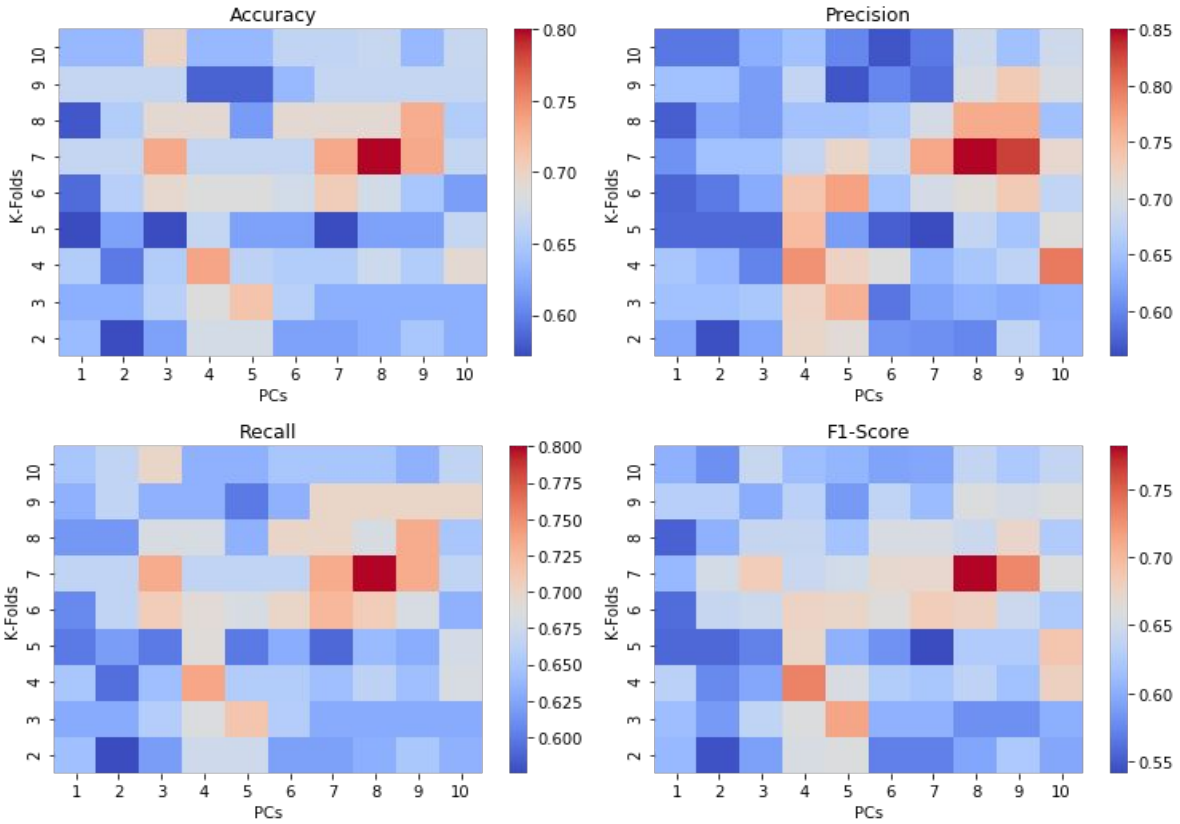
Training



Testing



Results - 310 Channels, 5 Classes



Conclusion

We can detect and classify single evoked potentials in a short time-window of data recorded from a 300+ channel microelectrode array placed subdurally on the spinal cord surface.

Initial hypothesis: extracting the top principal component (PC) from the input data will be sufficient to achieve a 50% classification accuracy rate via a multiclass linear discriminant analysis.

✗

- False for originally selected data with bad channels only removed based on impedance
- PC #6 necessary to classify data with accuracy > 50%

✓

- True for cleaned data where additional bad channels are removed
- PC #1 classified data with accuracy ~ 60%

Code: https://github.com/margotwagner/COGS260

Supplementary Slides

Analysis - Naive Bayes Classifier

- Bayes Theorem
- Calculate $P(x|C_j)$ from LDA
- Assume $P(C_j)$ from LDA
- Take class with highest conditional probability

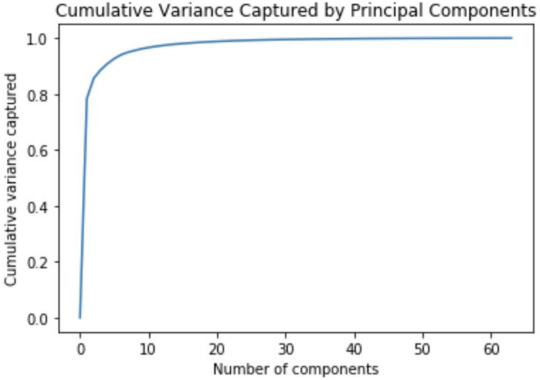
$$P(C_j|x) = \frac{P(x|C_j)P(C_j)}{P(x)}$$

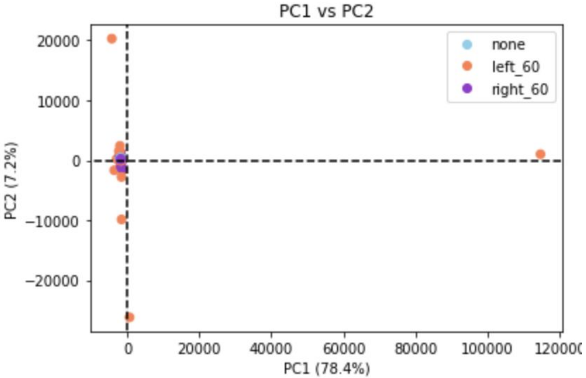
$$P(x) = \sum_{j=1}^{2} P(x|C_j)P(C_j)$$

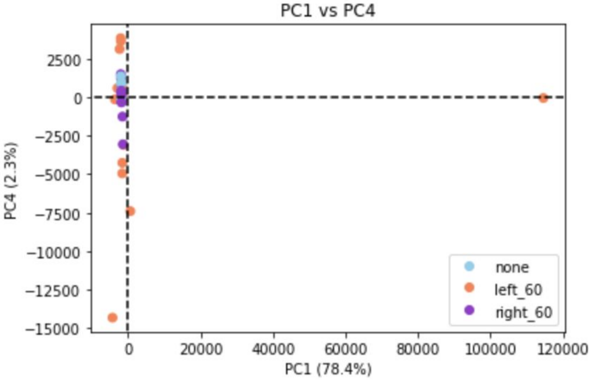
$$\text{Decide} \begin{cases} C_1 & \text{if} \quad P(C_1|x) > P(C_2|x) \\ C_2 & otherwise \end{cases}$$

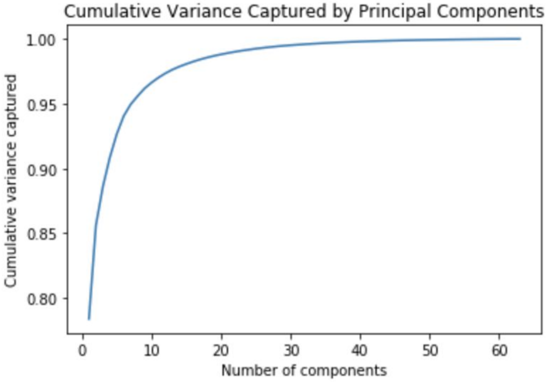
https://web.fe.up.pt/~jsc/ML/lectures/lecture03.pdf

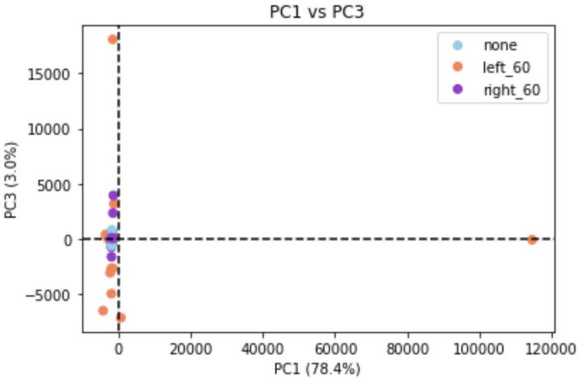
Additional Slides - PCA 353 Channels

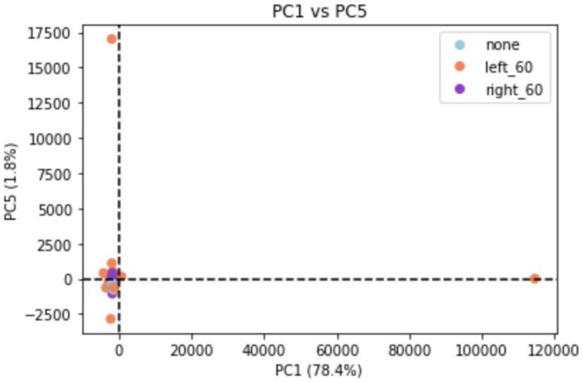




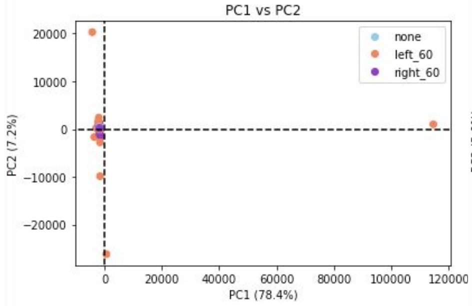


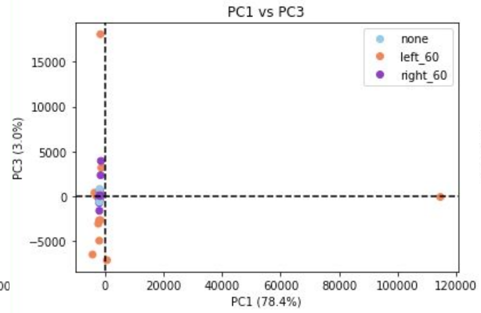


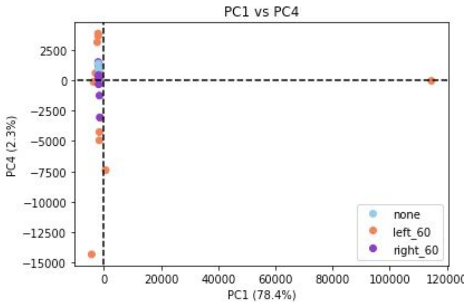


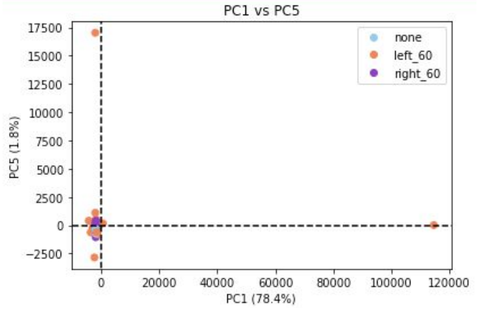


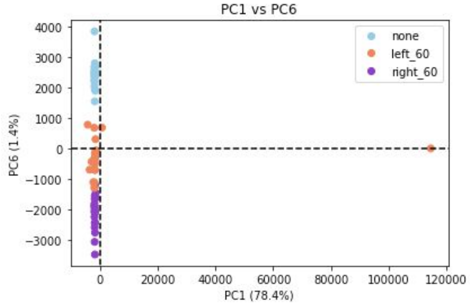
Additional Slides - PCA 353 Channels PC1

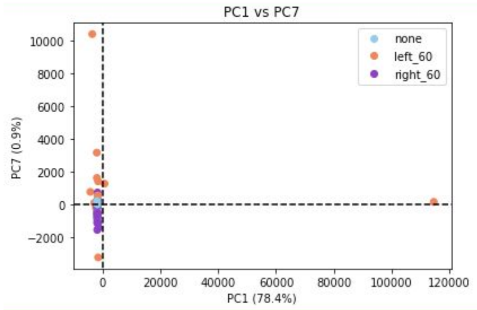


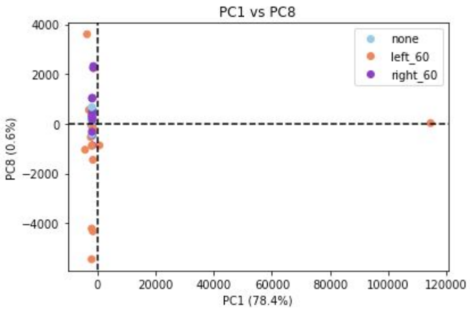


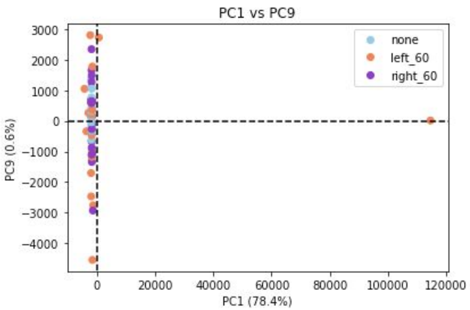


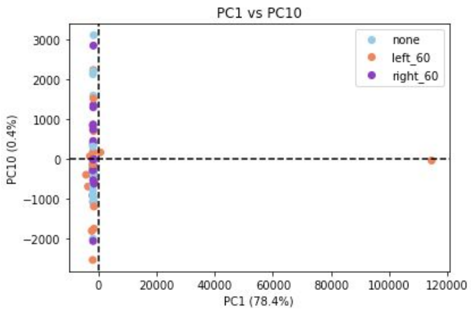




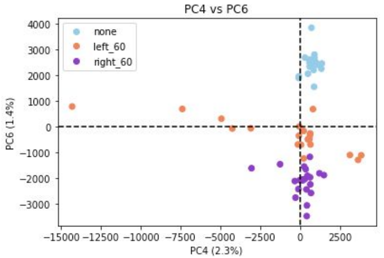


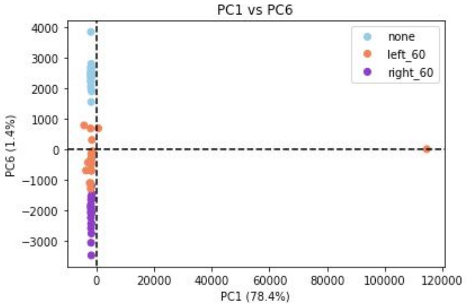


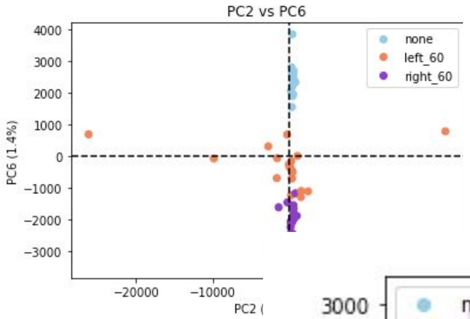


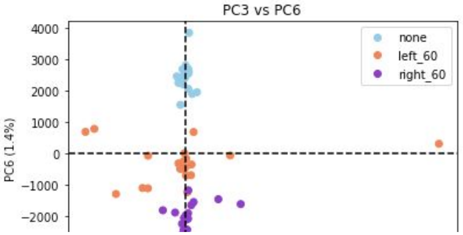


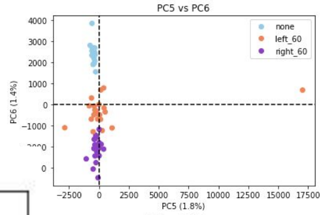
Additional Slides - PCA 353 Channels PC6

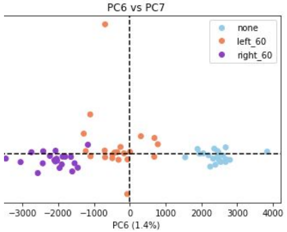


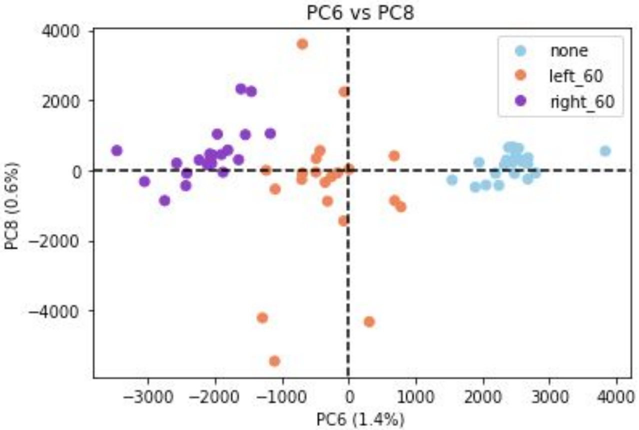


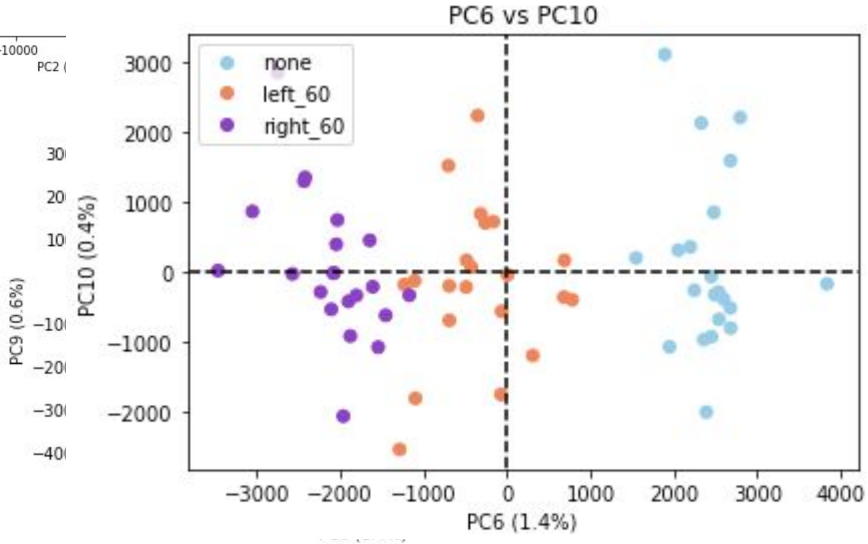




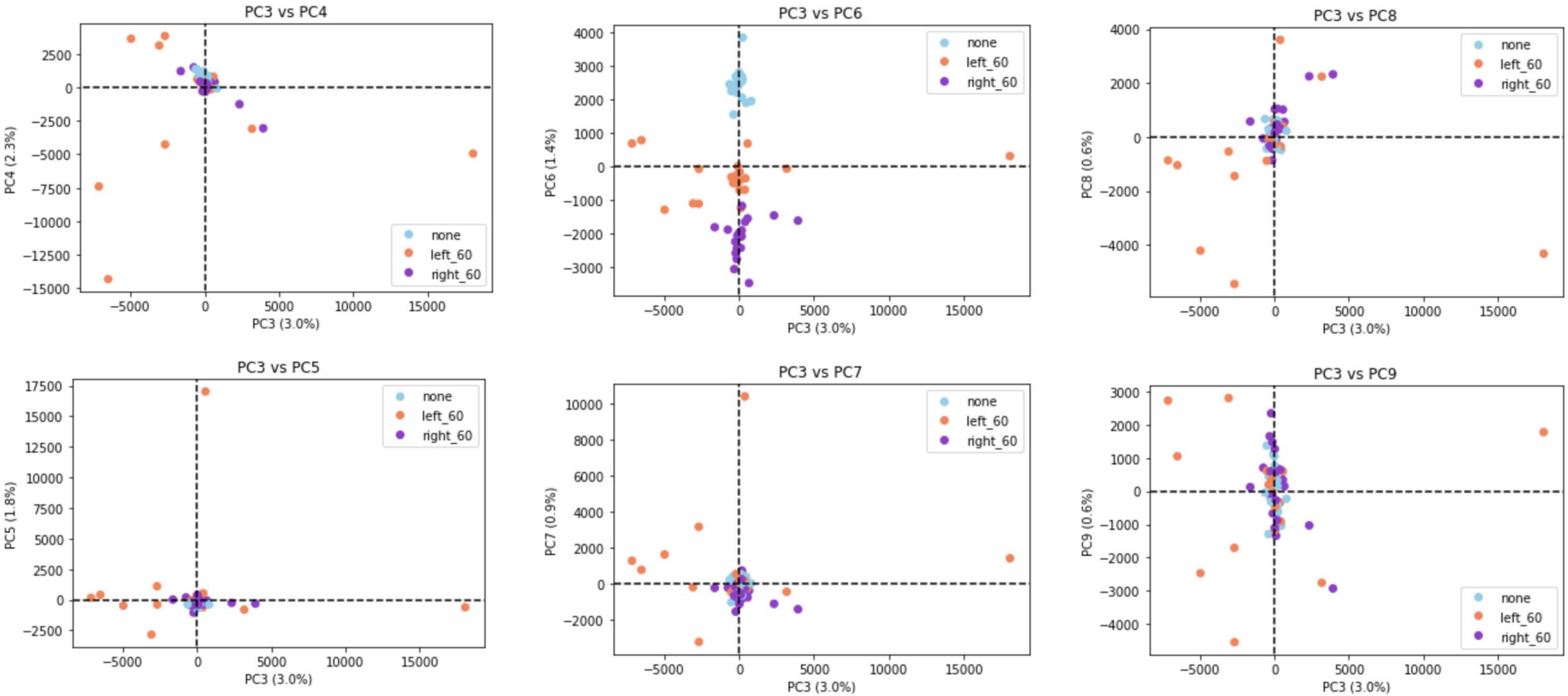




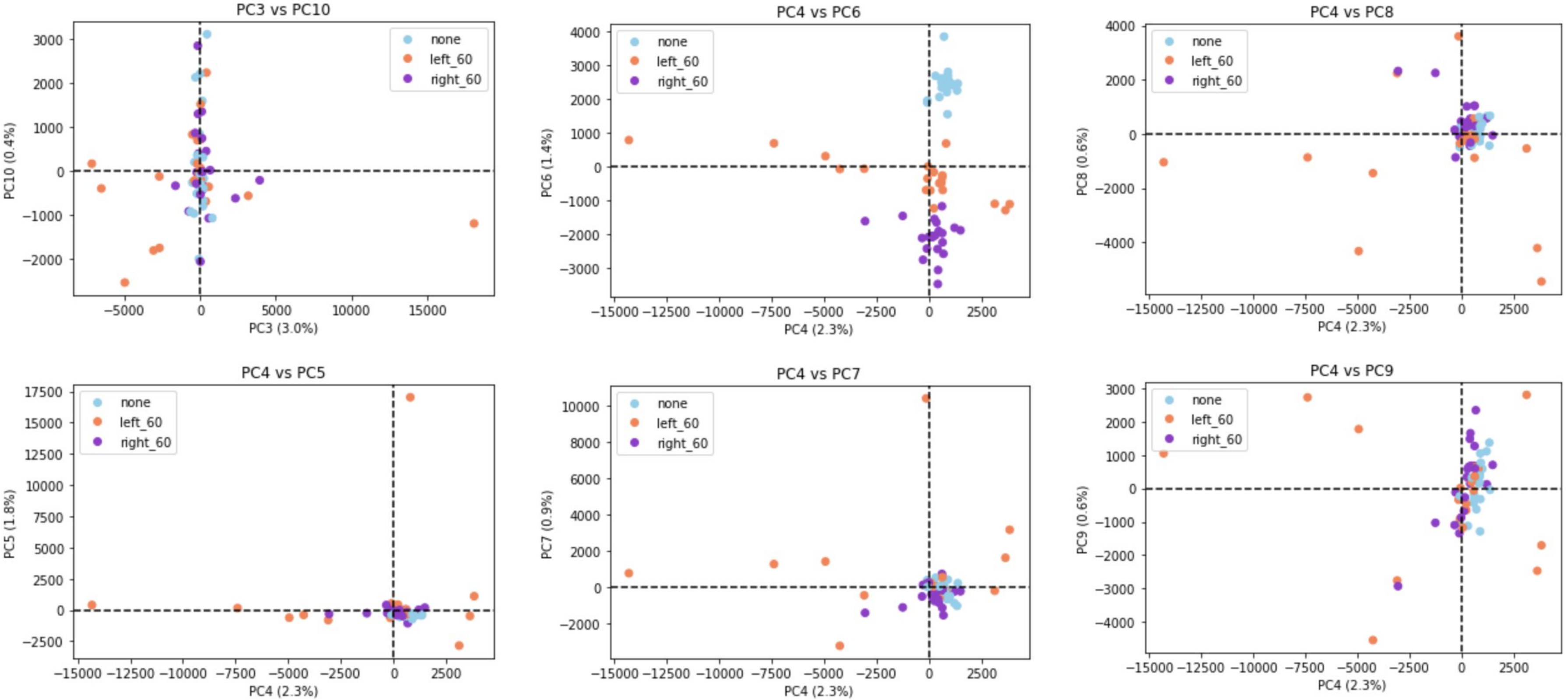




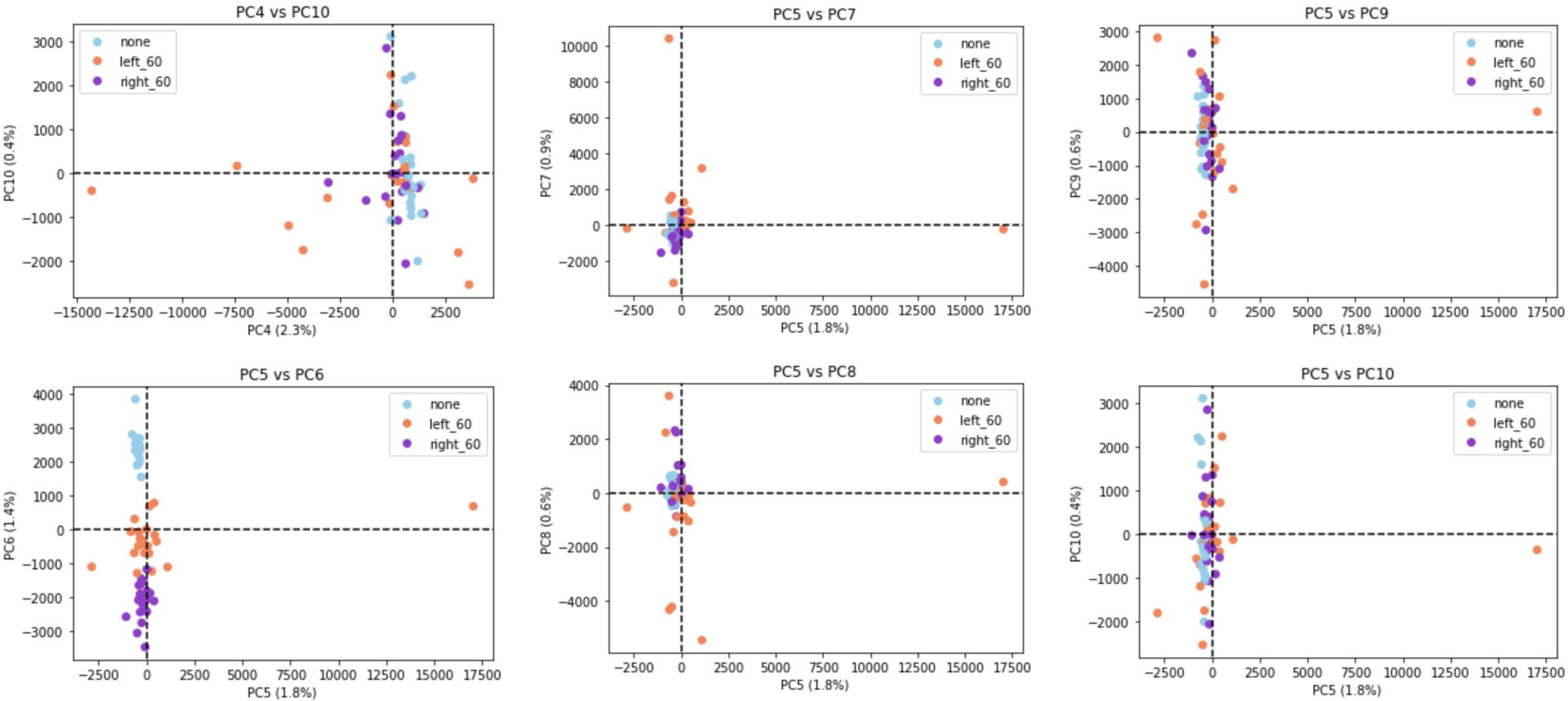
Additional Slides - PCA 353 Channels Other



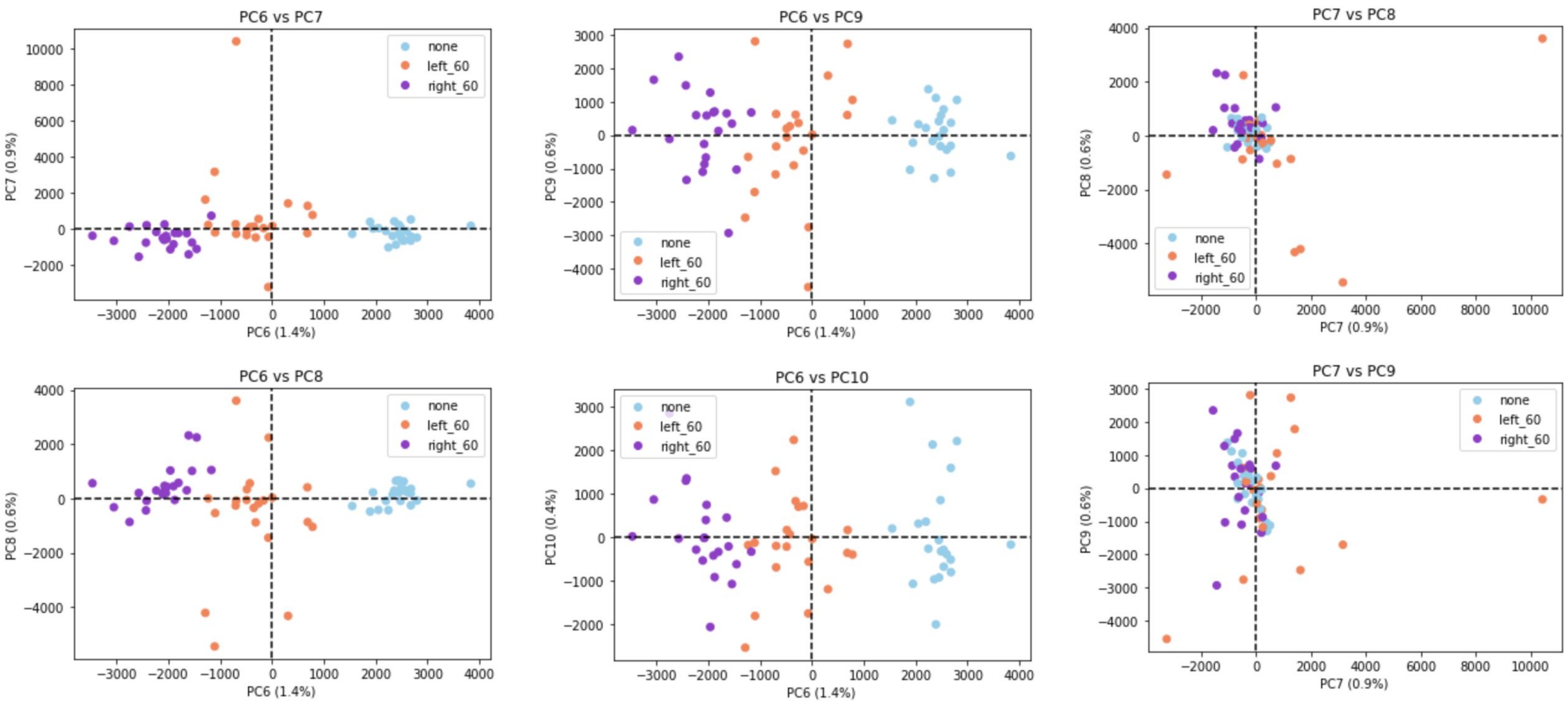
Additional Slides - PCA 353 Channels Other



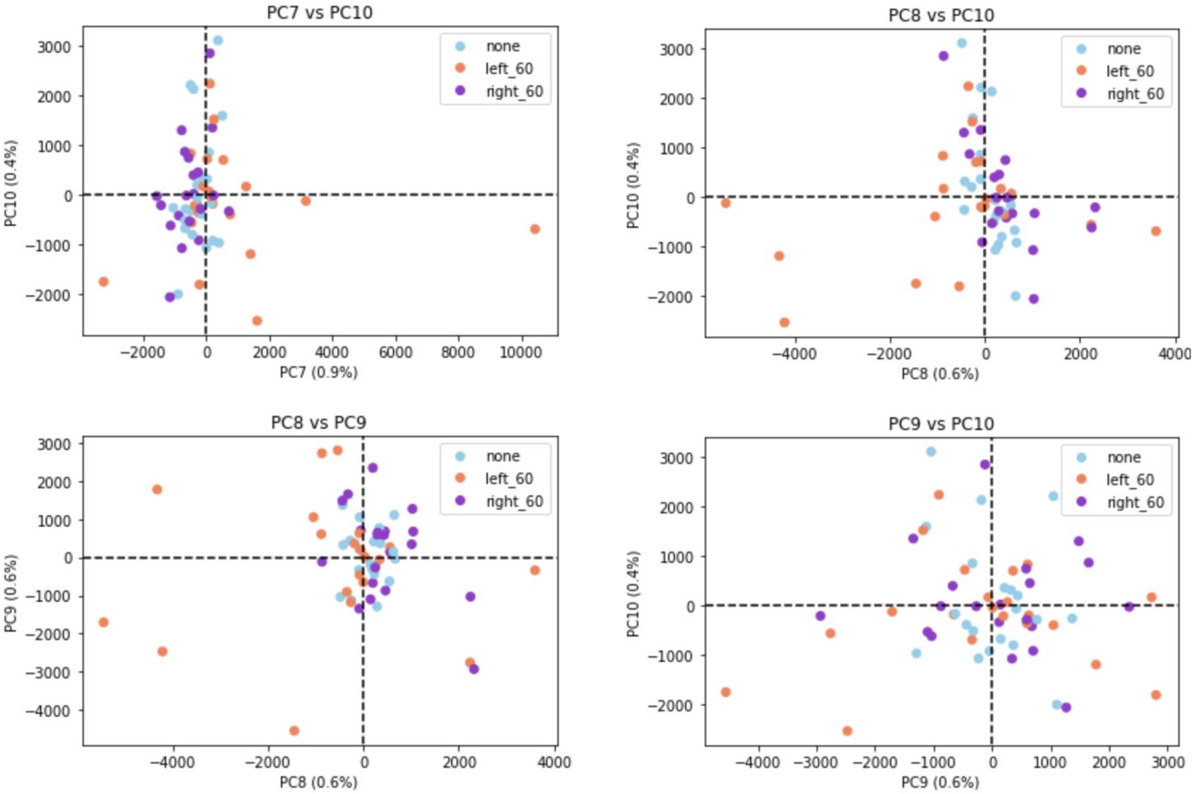
Additional Slides - PCA 353 Channels Other



Additional Slides - PCA 353 Channels Other

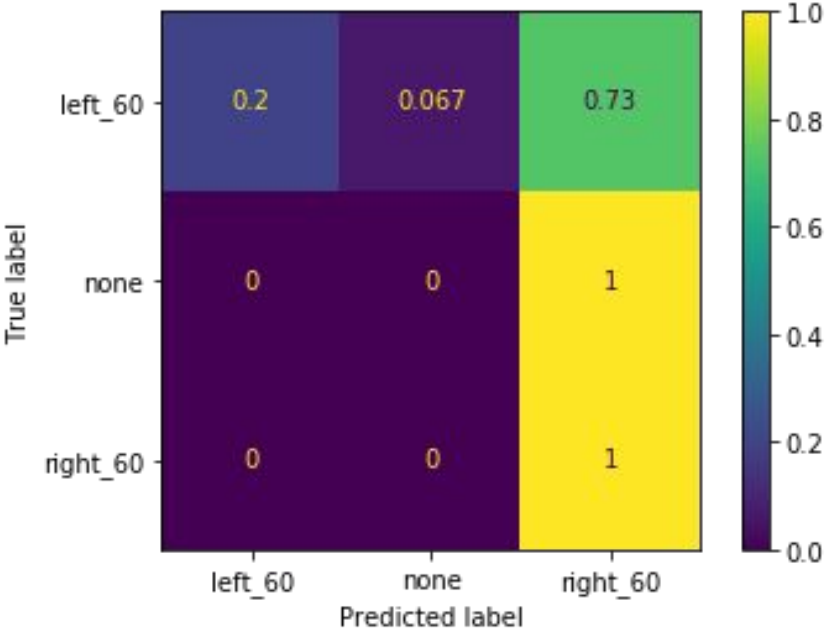


Additional Slides - PCA 353 Channels Other

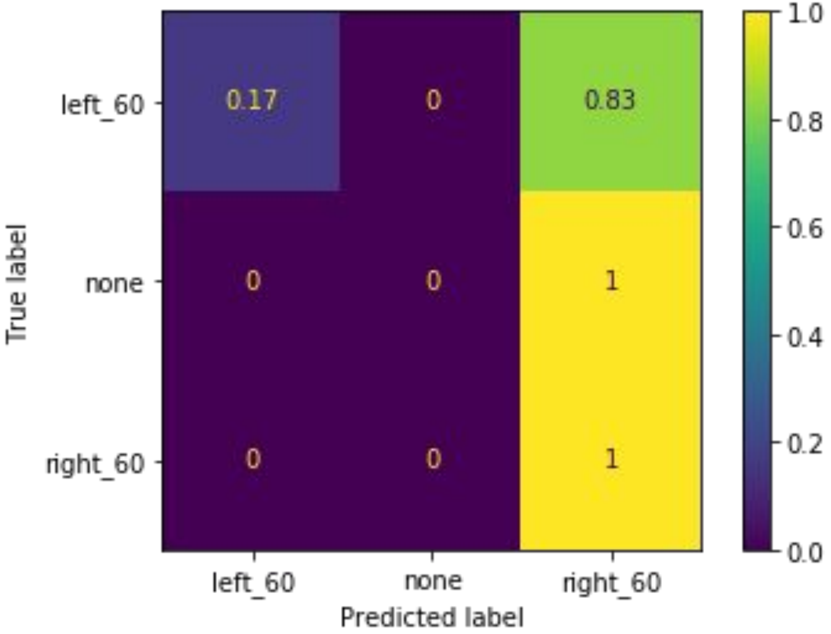


Confusion matrix - 353 channels

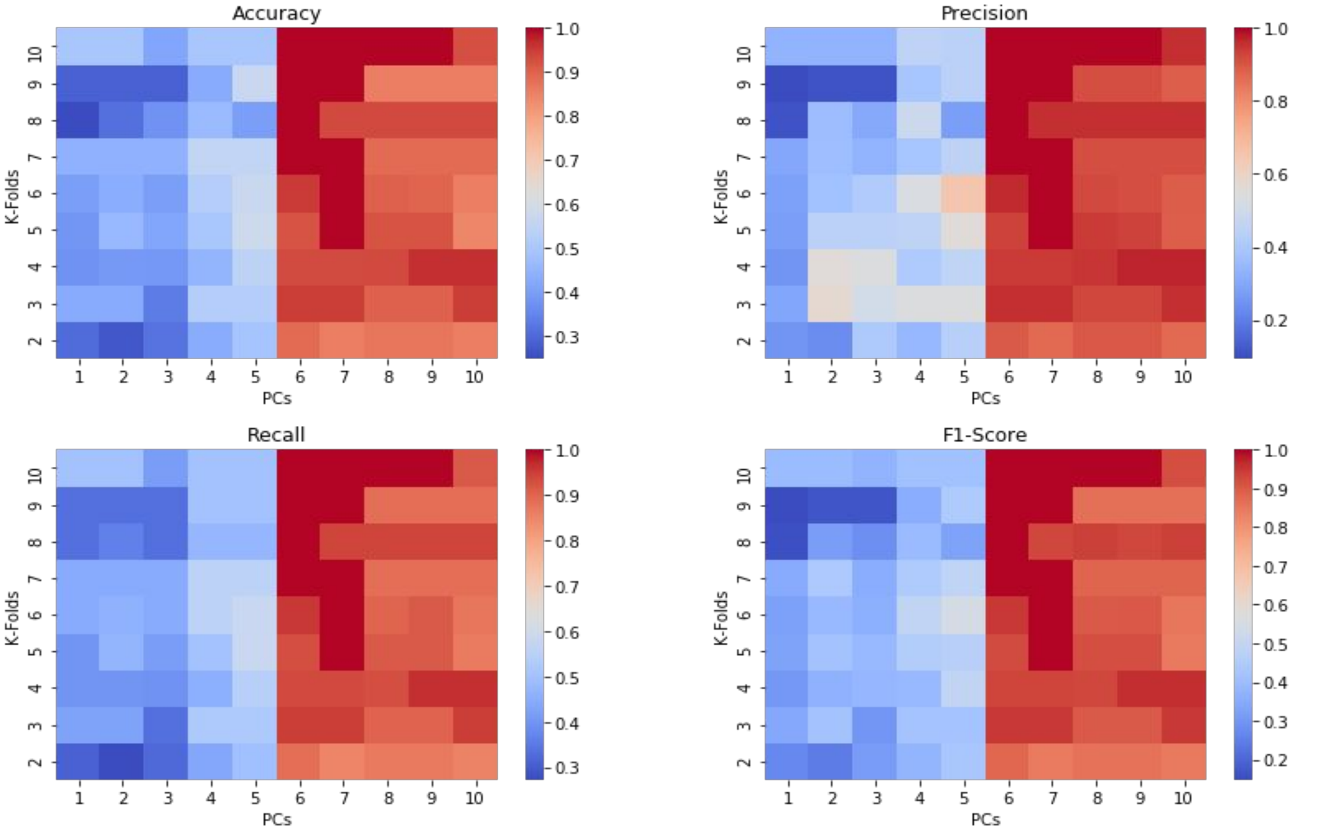
Training



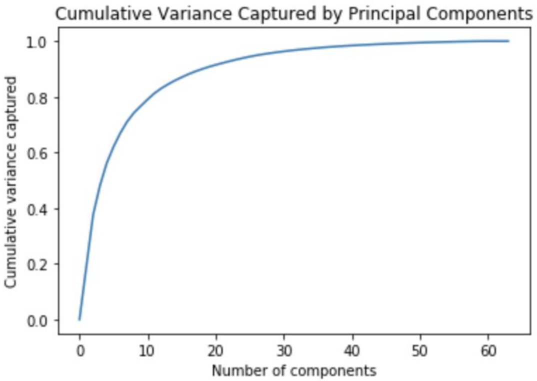
Testing

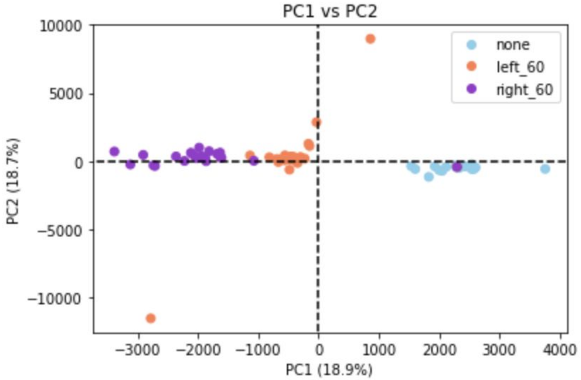


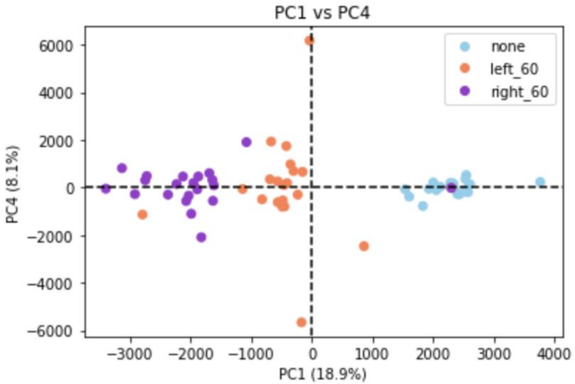
Heatmaps - 353 Channels

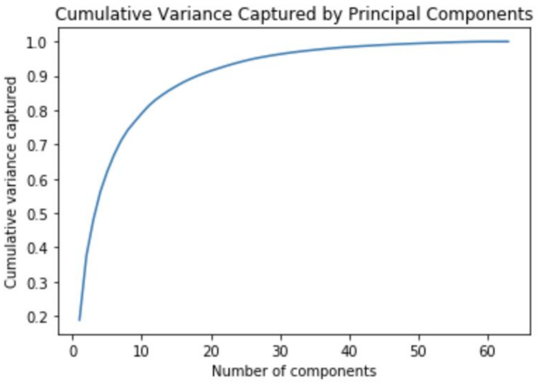


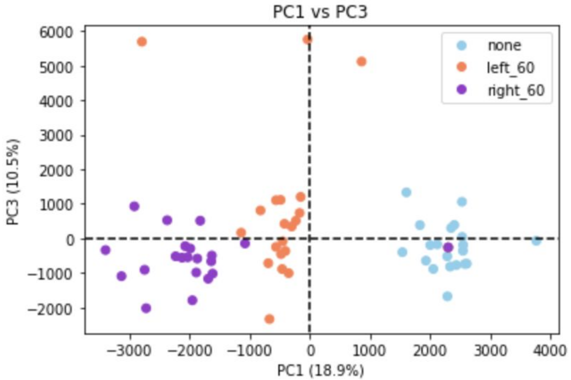
Additional Slides - PCA 310 Channels 3 Conditions

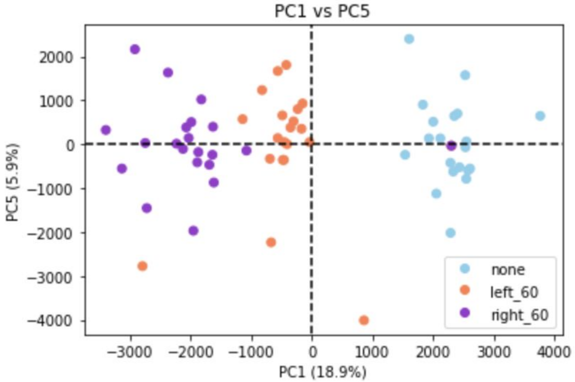




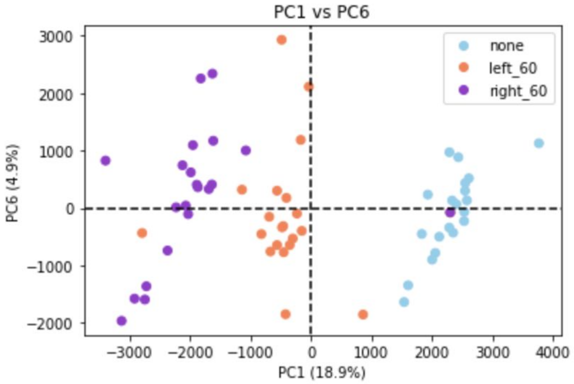


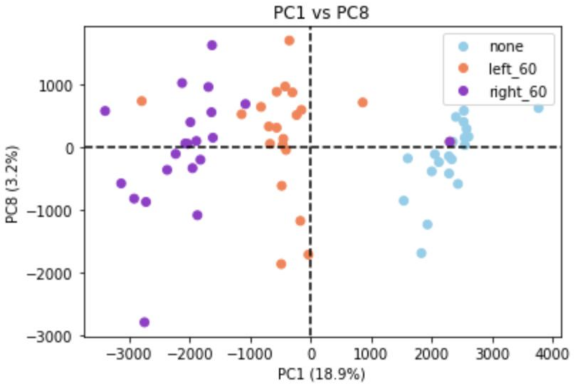


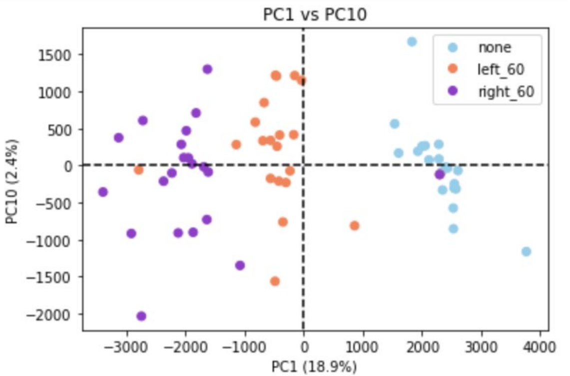


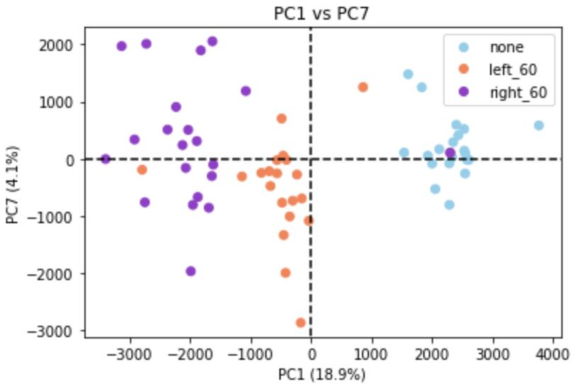


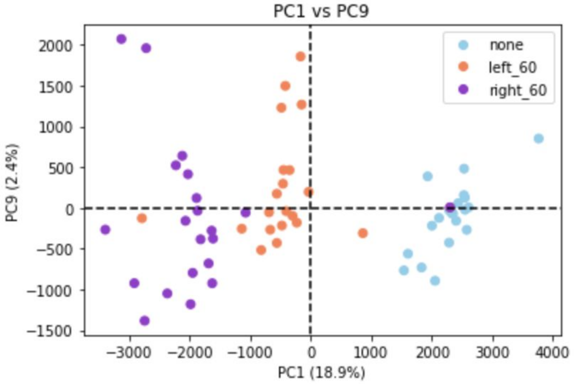
Additional Slides - PCA 310 Channels 3 Conditions

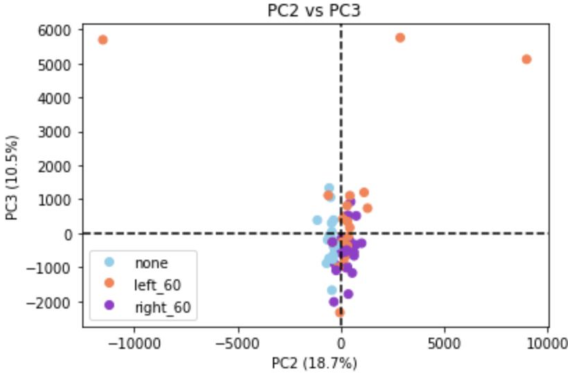




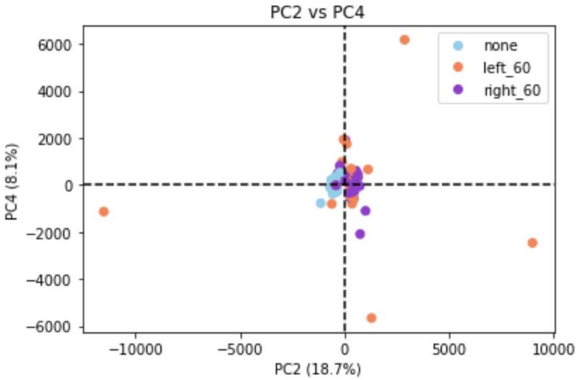


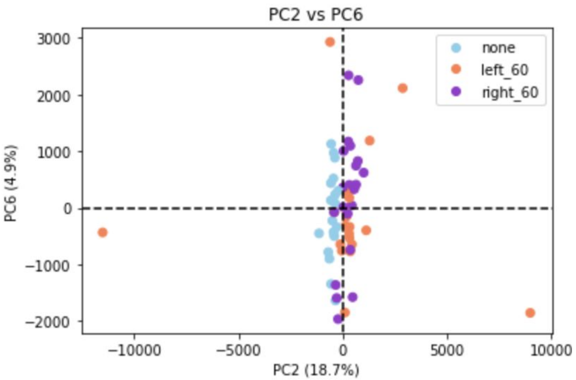


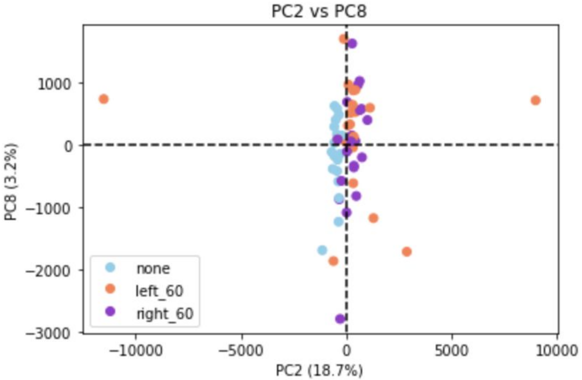


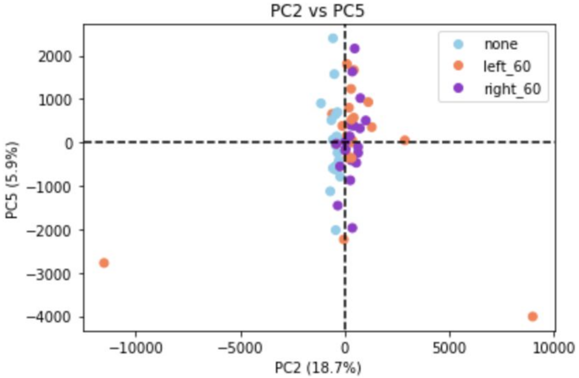


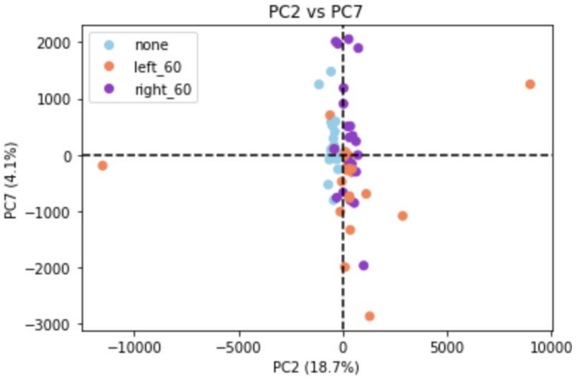
Additional Slides - PCA 310 Channels 3 Conditions

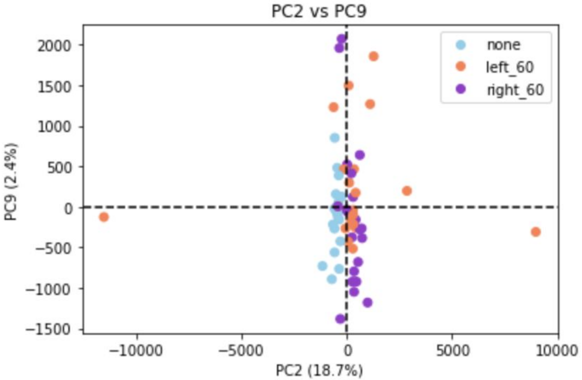




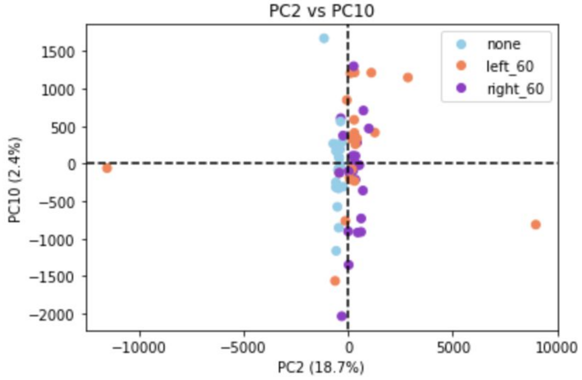


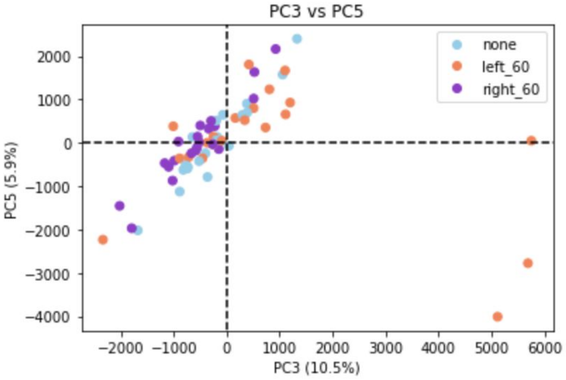


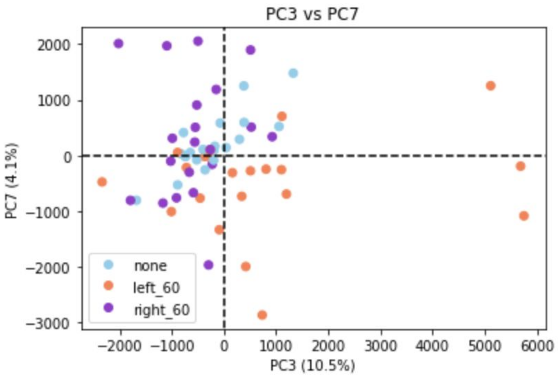


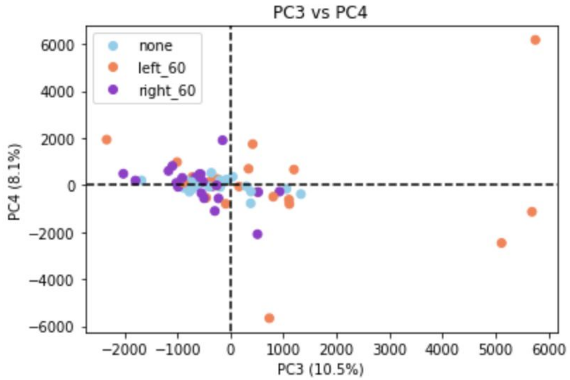


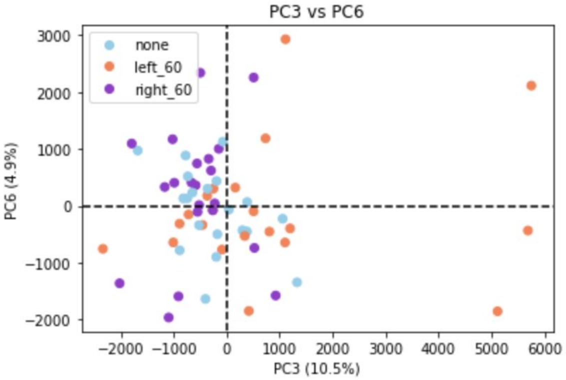
Additional Slides - PCA 310 Channels 3 Conditions

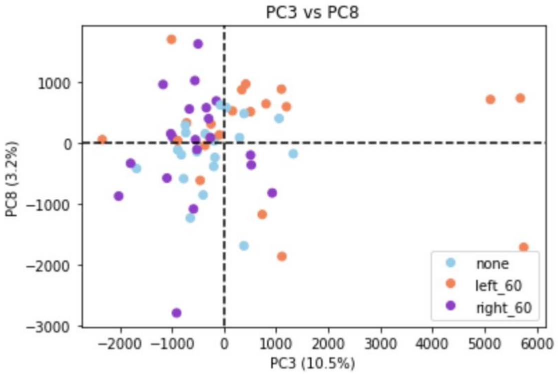




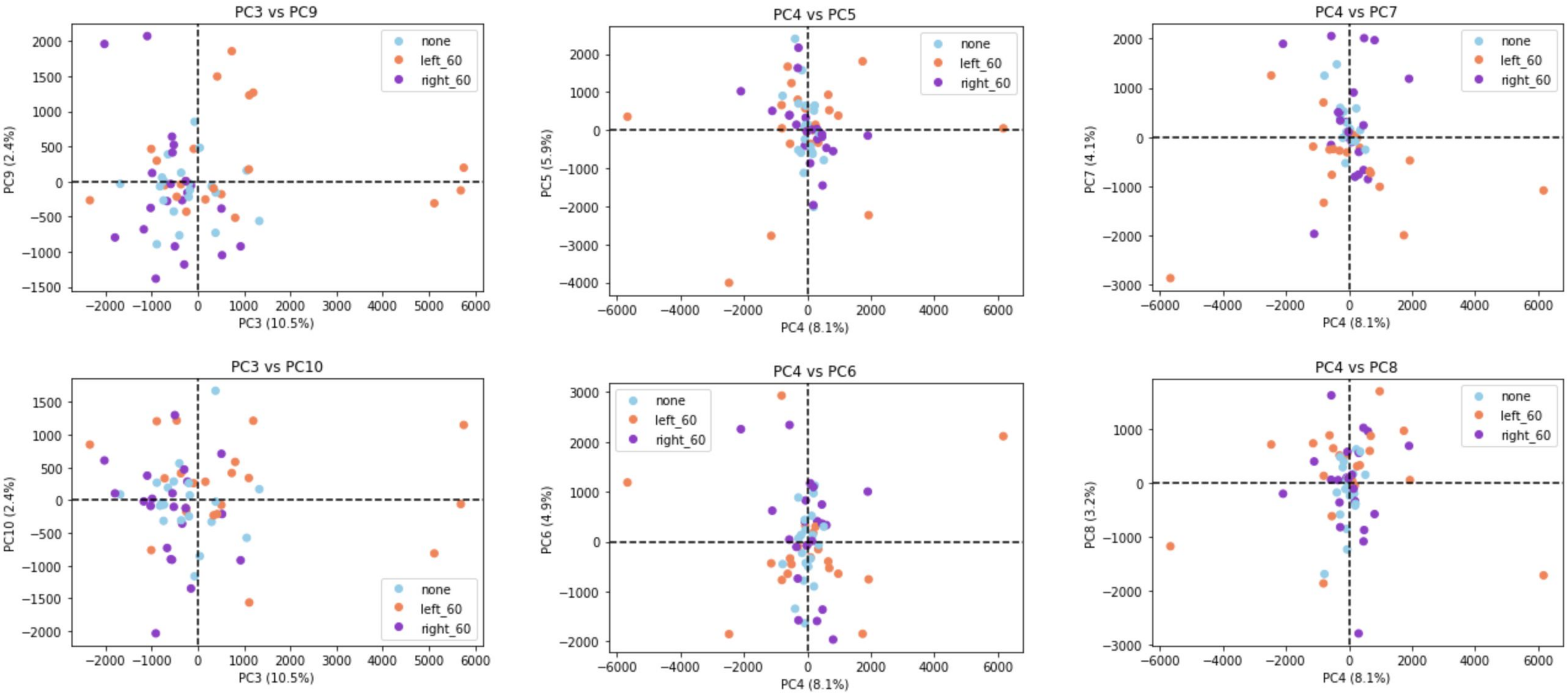




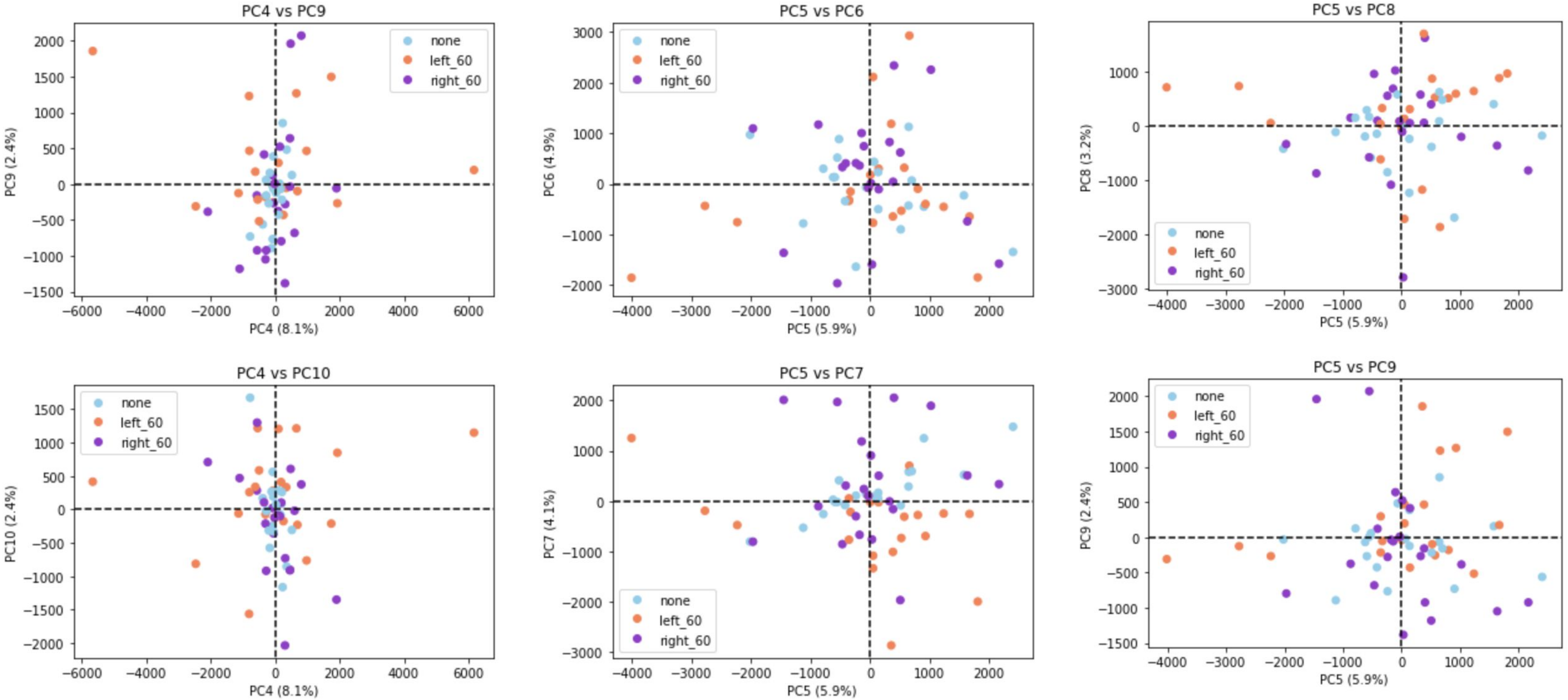




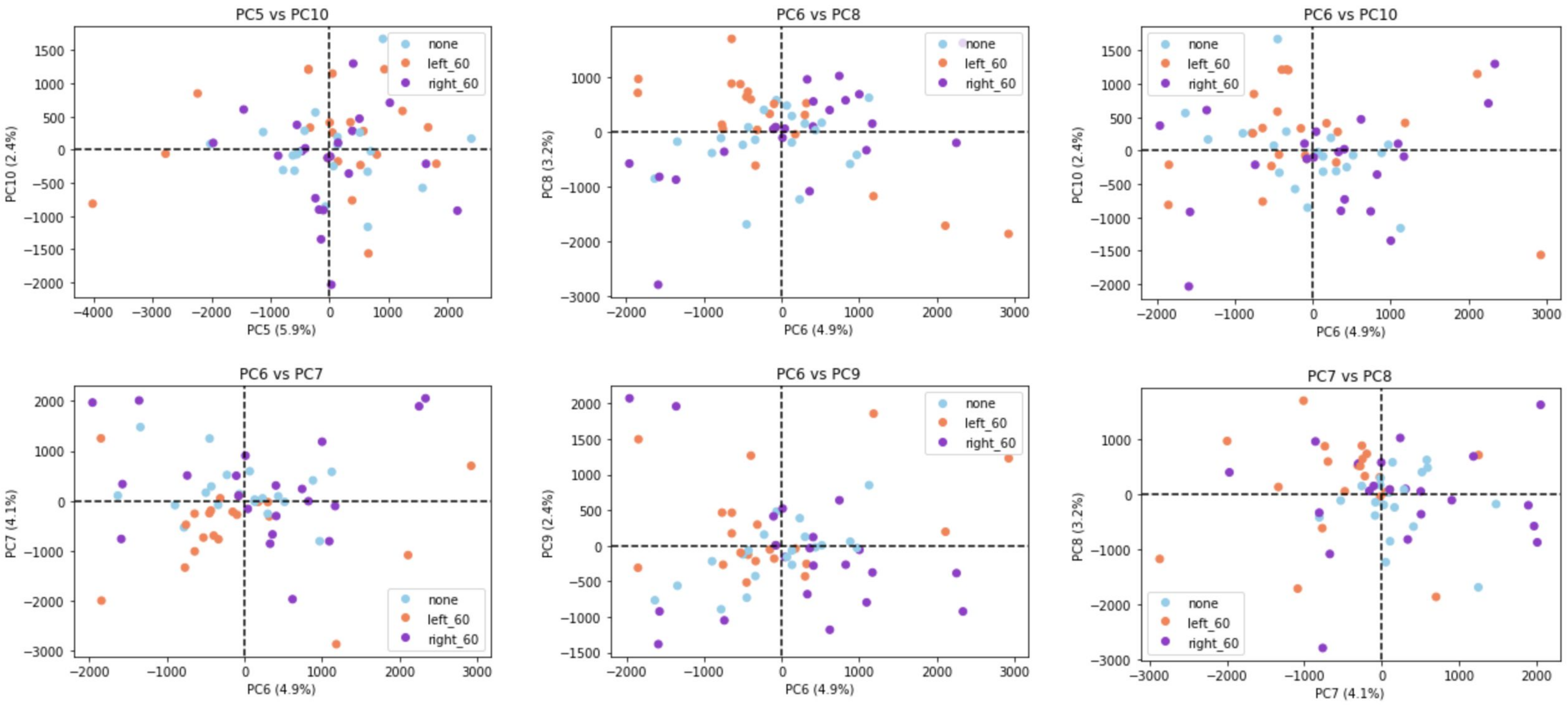
Additional Slides - PCA 310 Channels 3 Conditions



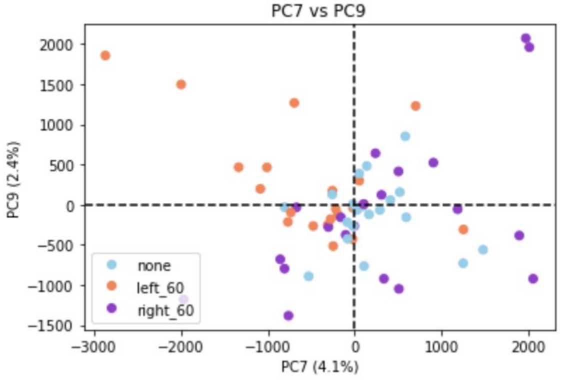
Additional Slides - PCA 310 Channels 3 Conditions

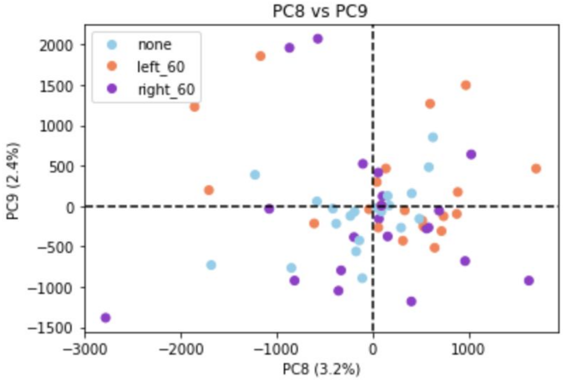


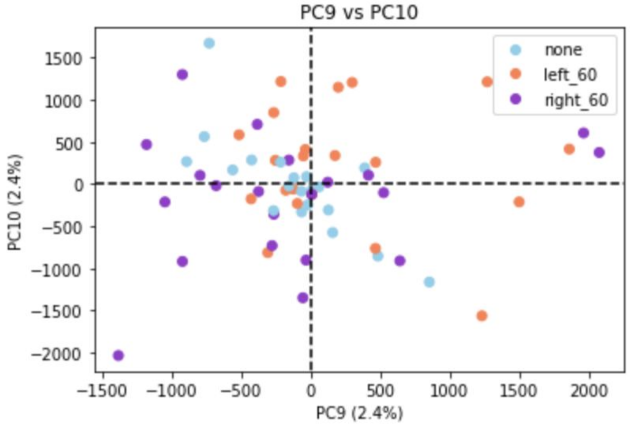
Additional Slides - PCA 310 Channels 3 Conditions

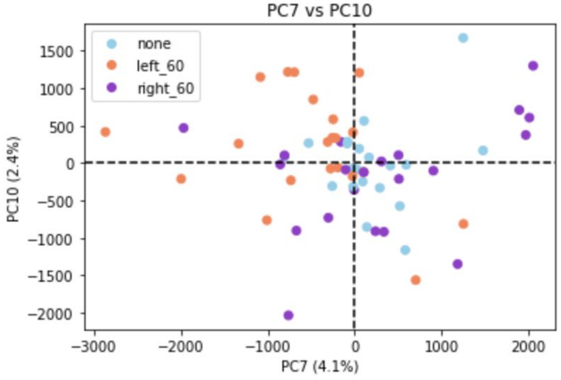


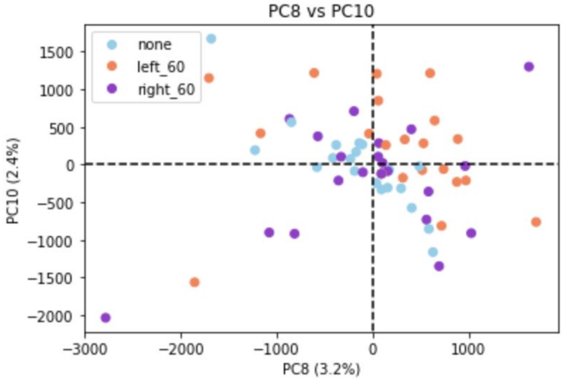
Additional Slides - PCA 310 Channels 3 Conditions





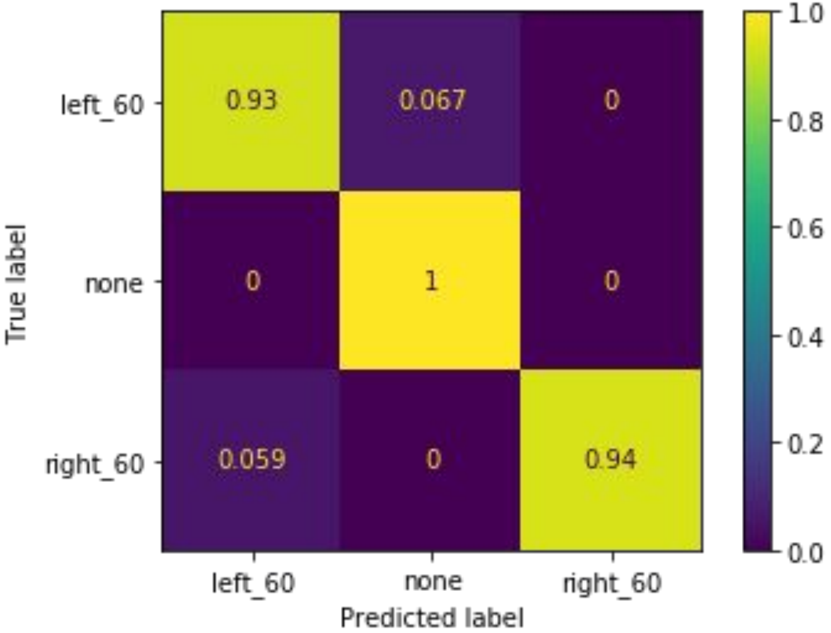




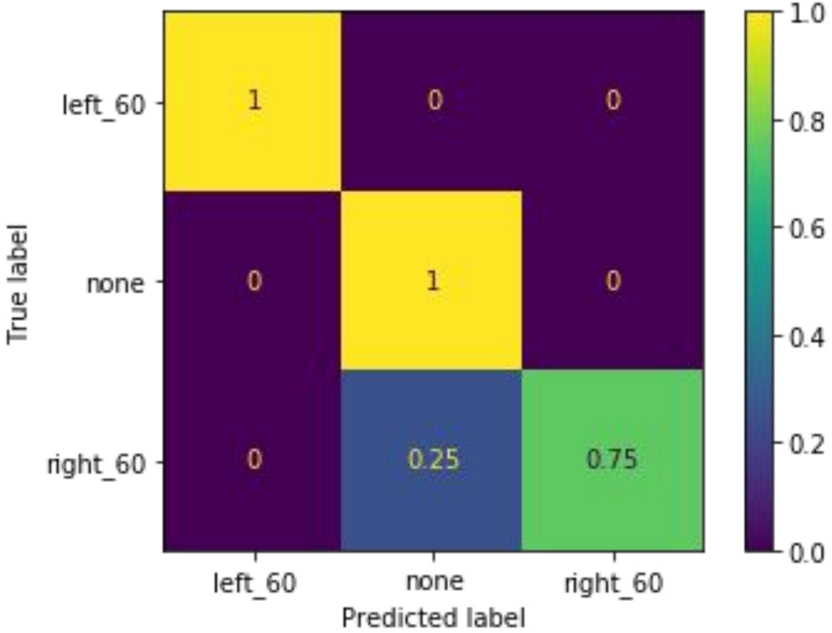


Confusion matrix - 310 channels, 3 groups

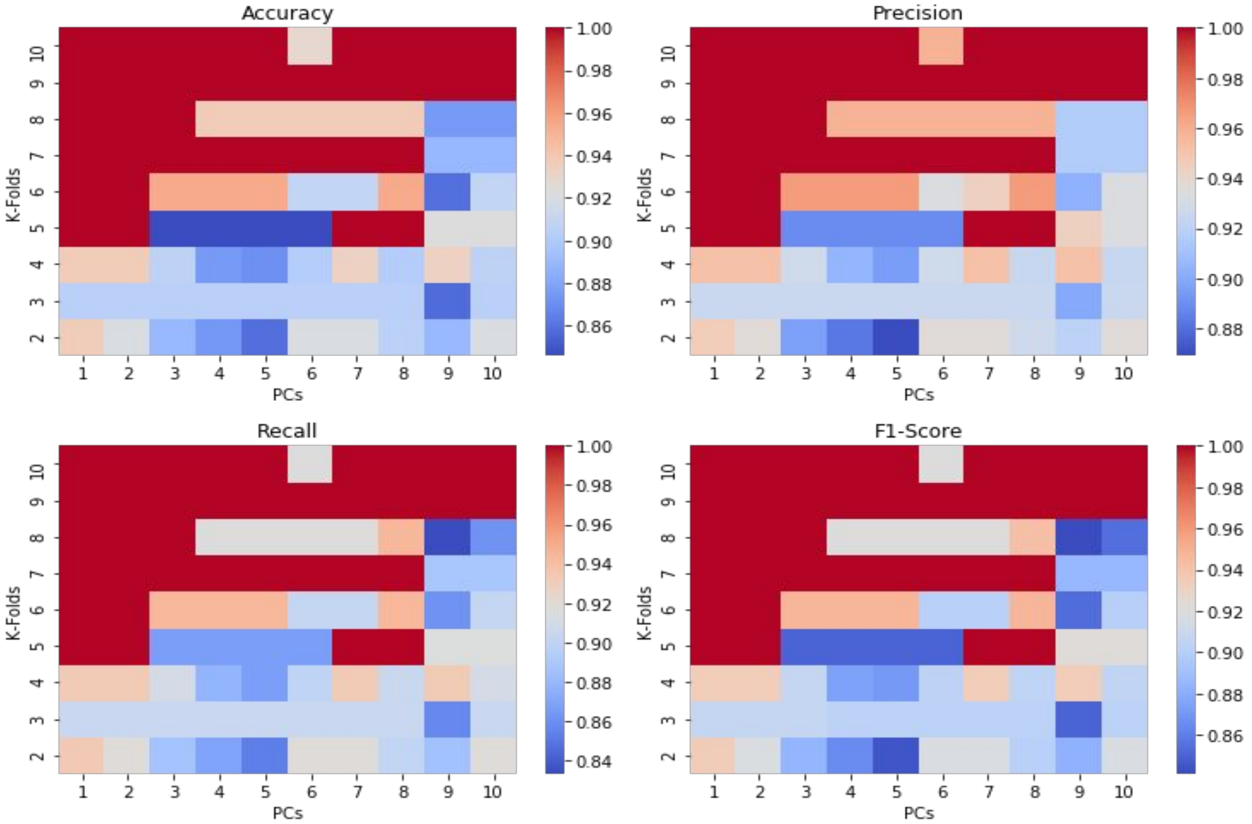
Training



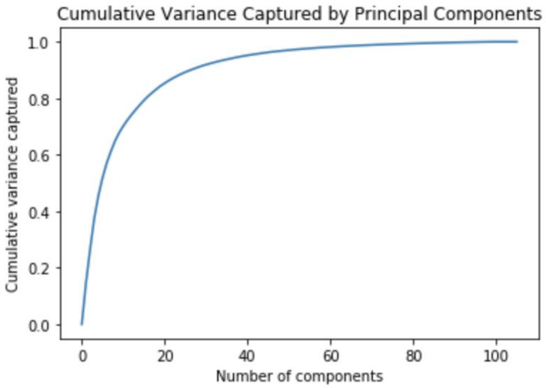
Testing

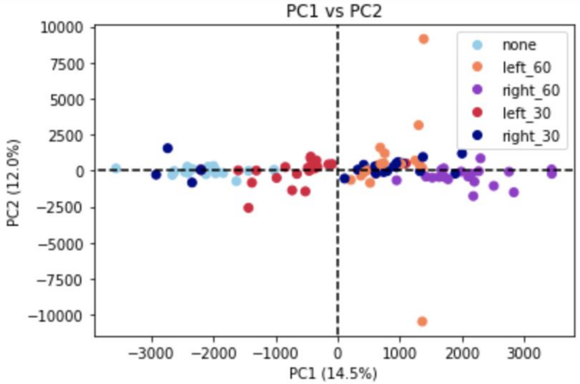


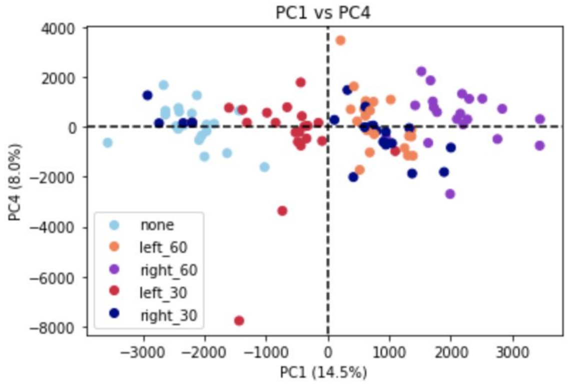
Heatmaps - 310 Channels, 3 groups

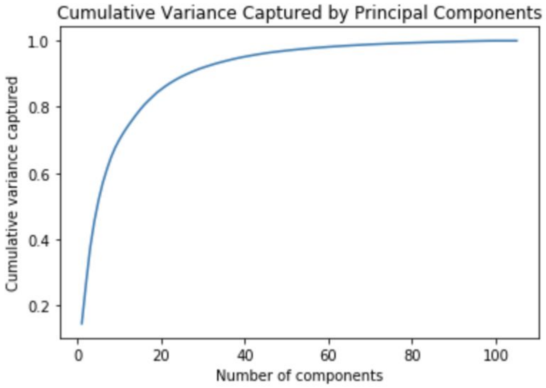


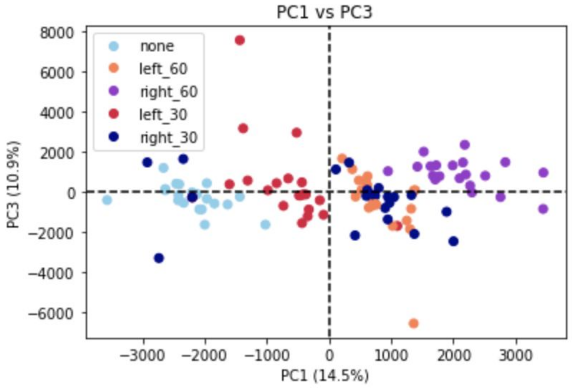
Additional Slides - PCA 310 Channels 5 Conditions

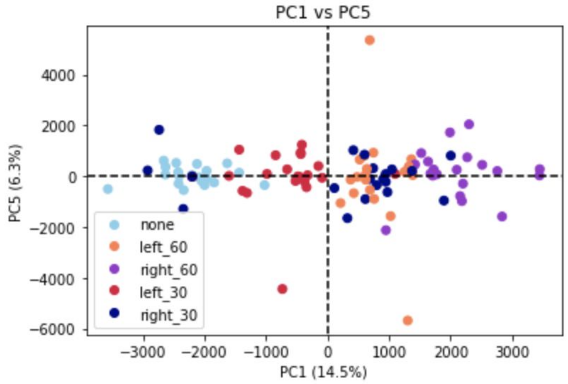




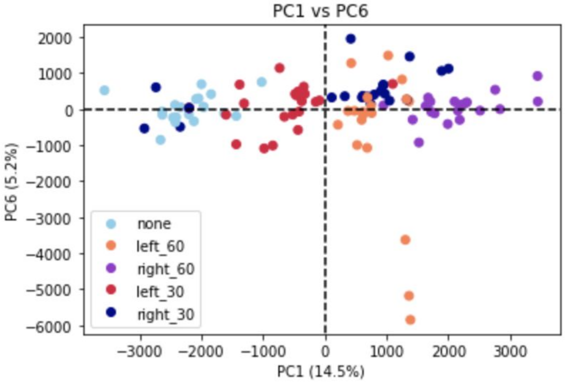


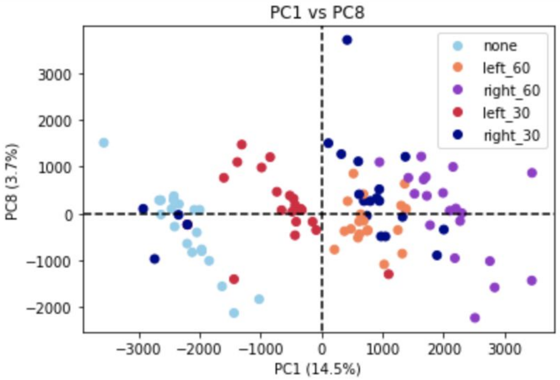


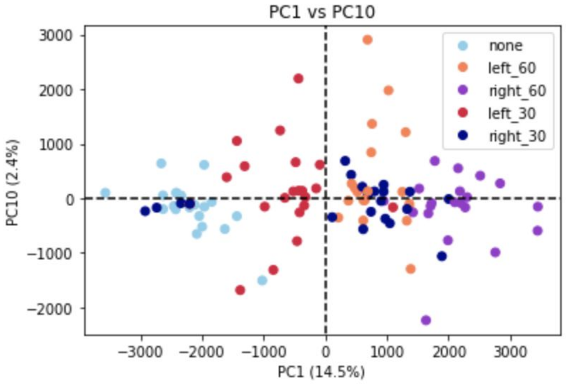


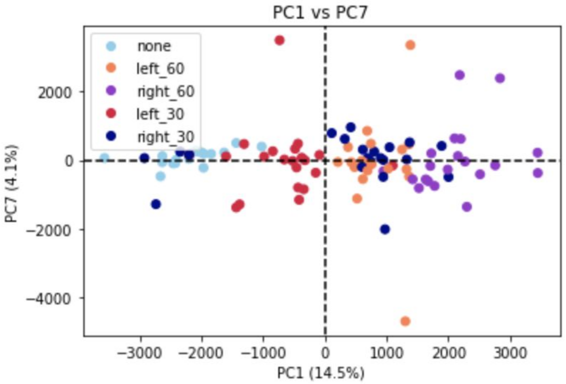


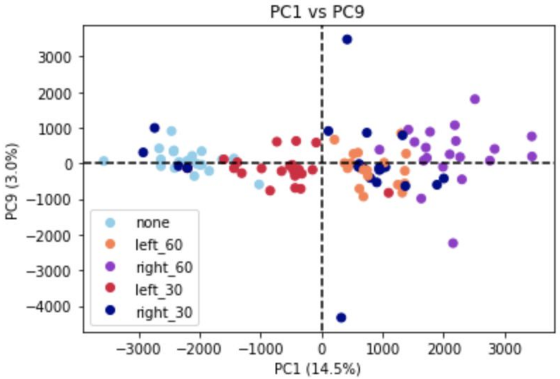
Additional Slides - PCA 310 Channels 5 Conditions

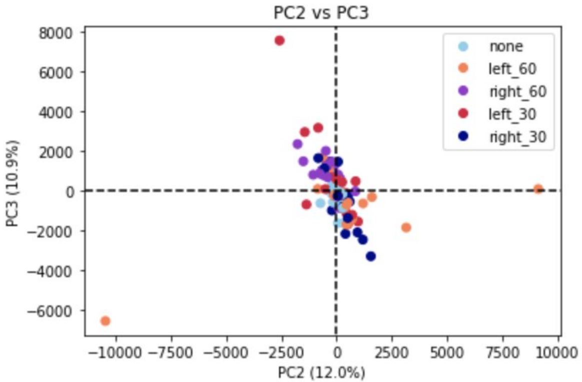




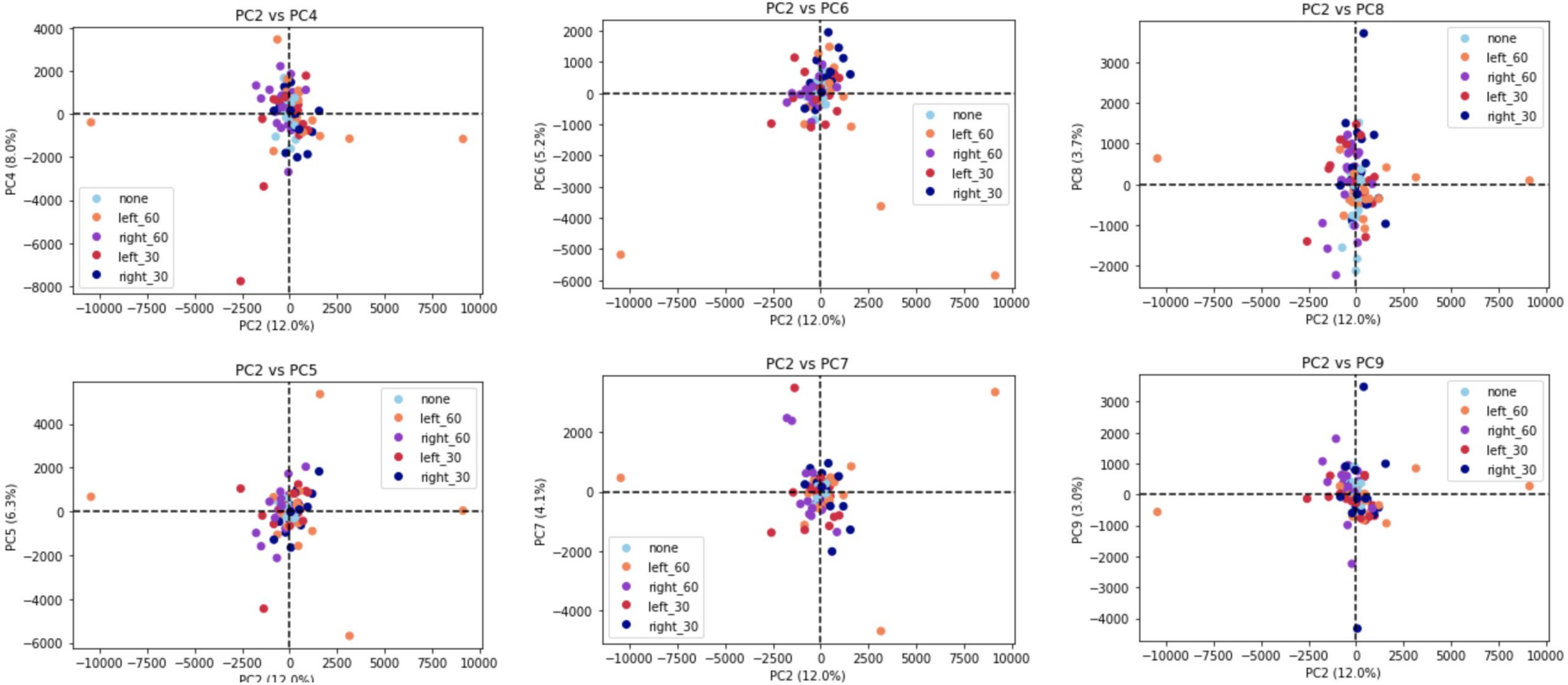




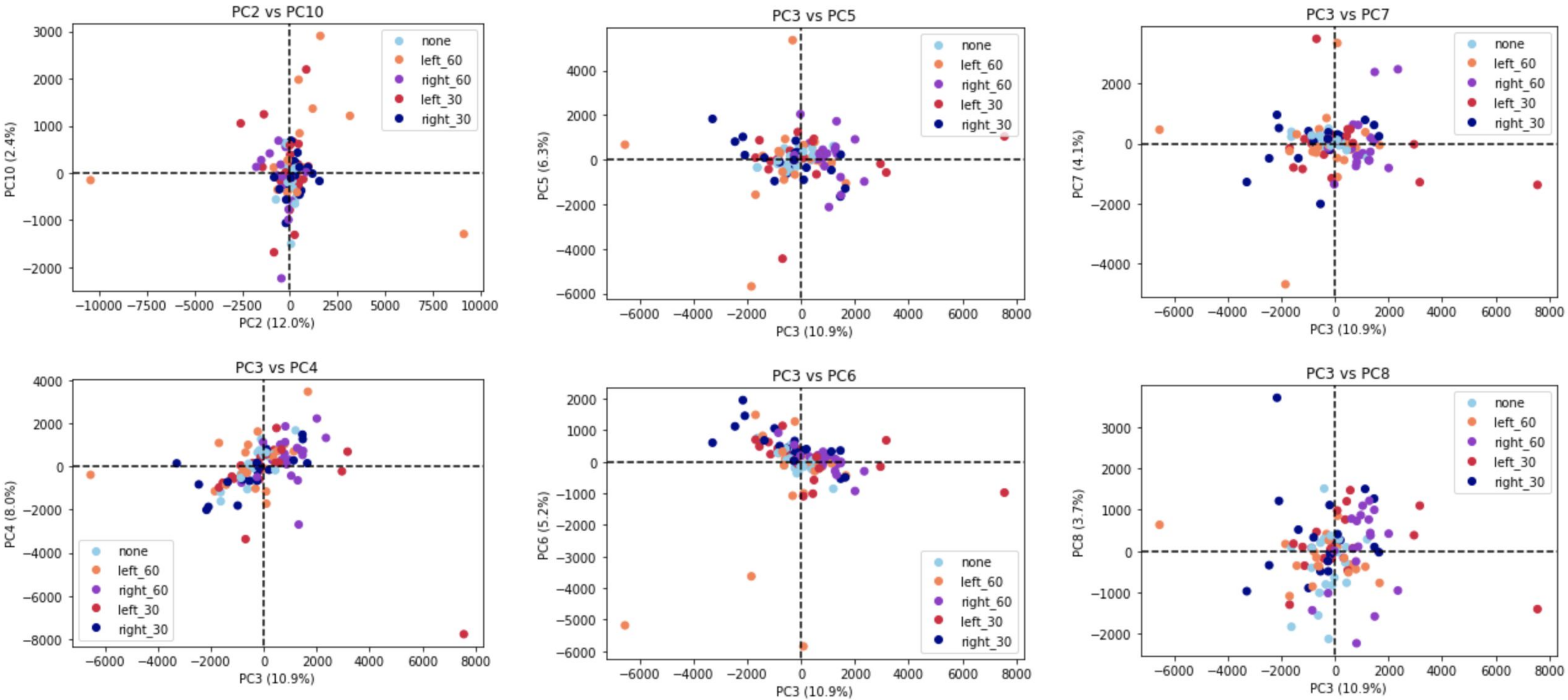




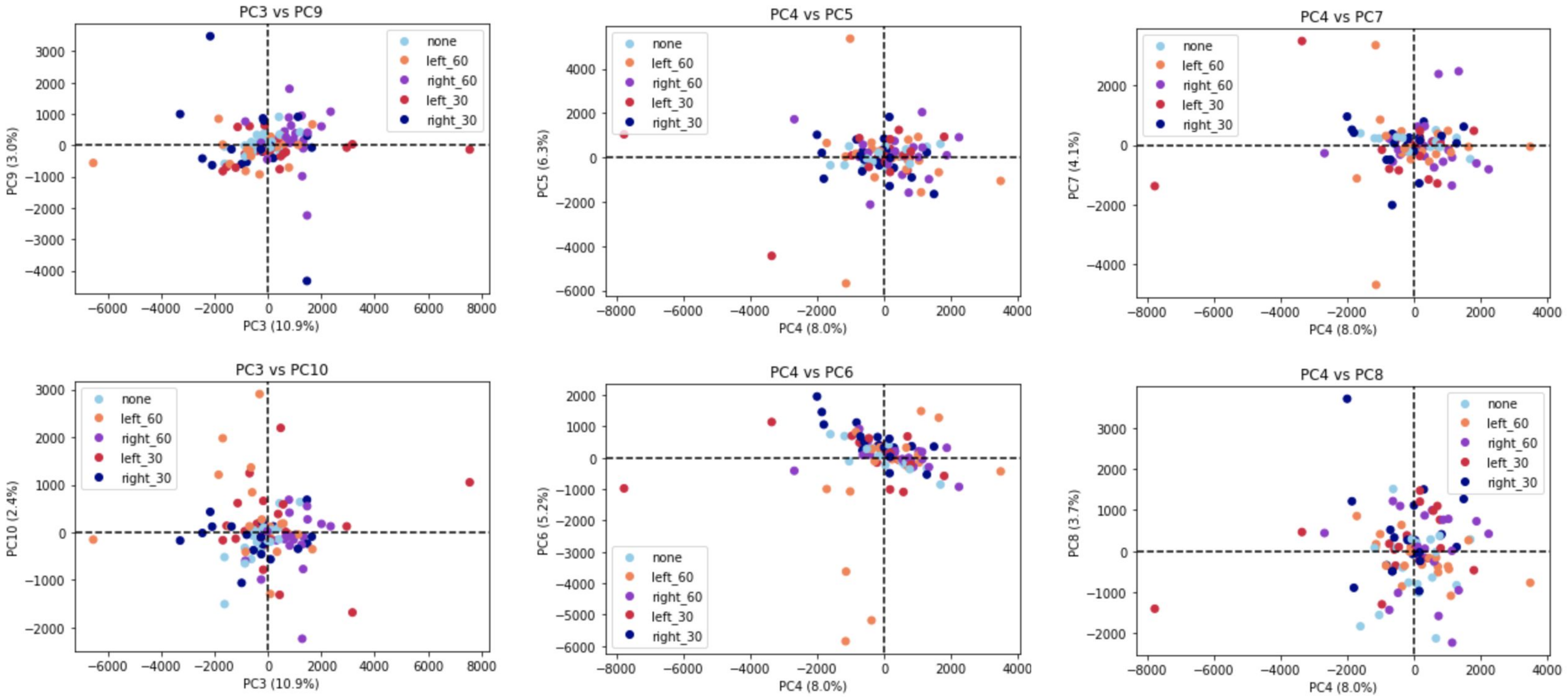
Additional Slides - PCA 310 Channels 5 Conditions



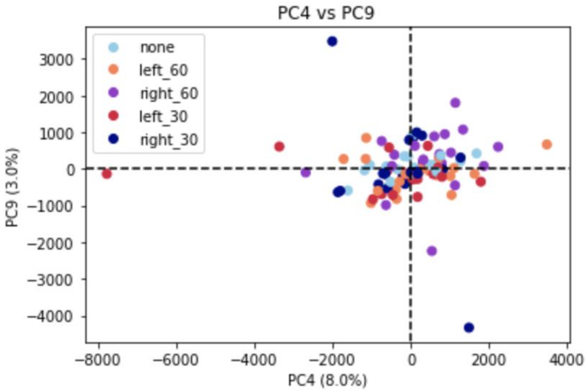
Additional Slides - PCA 310 Channels 5 Conditions

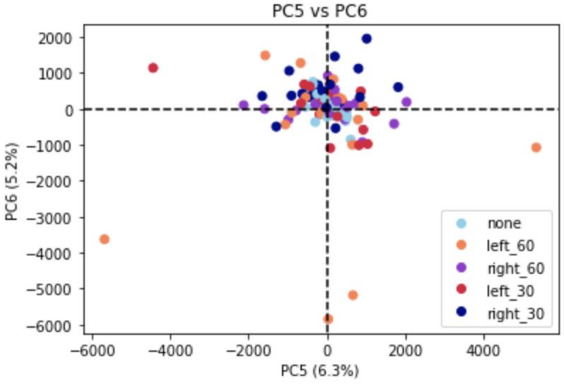


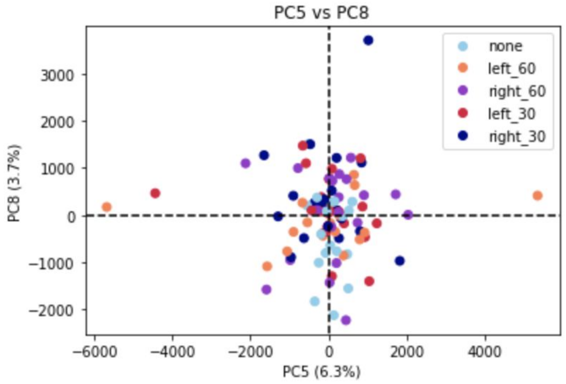
Additional Slides - PCA 310 Channels 5 Conditions

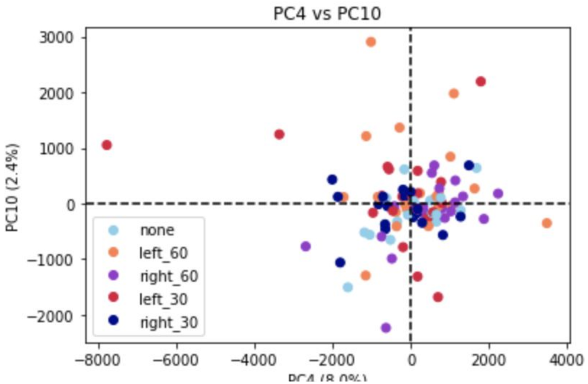


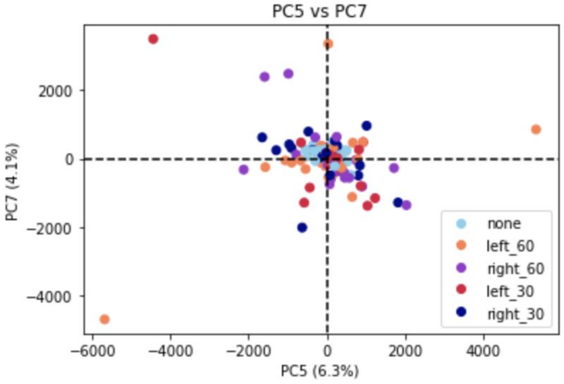
Additional Slides - PCA 310 Channels 5 Conditions

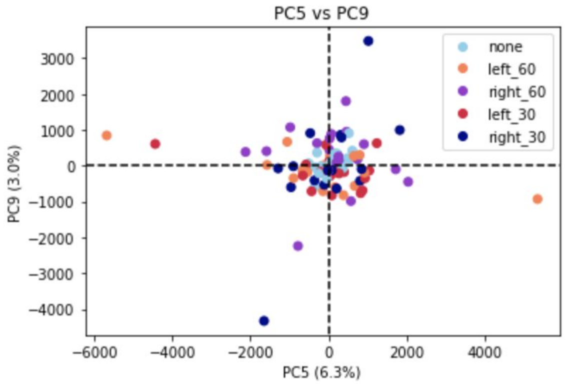




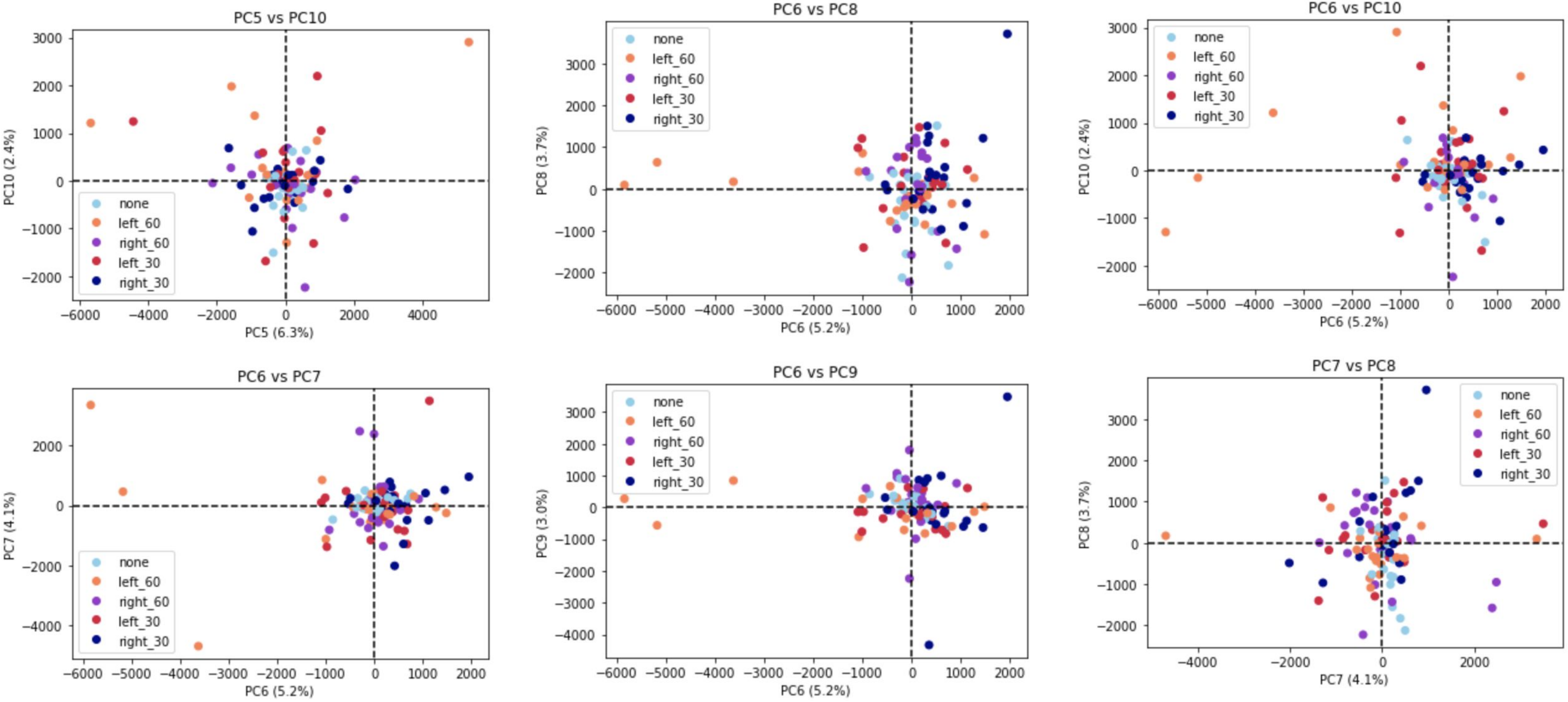




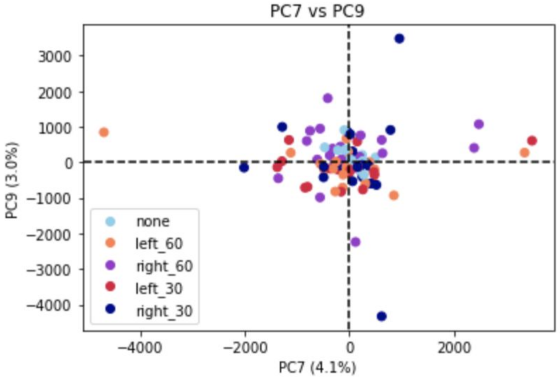


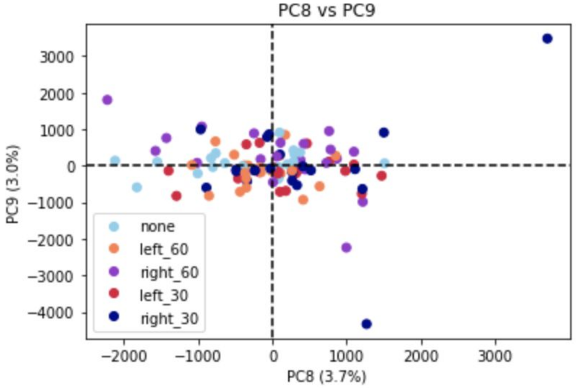


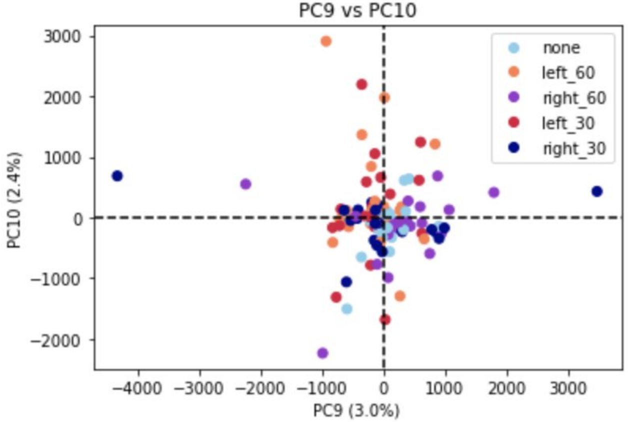
Additional Slides - PCA 310 Channels 5 Conditions

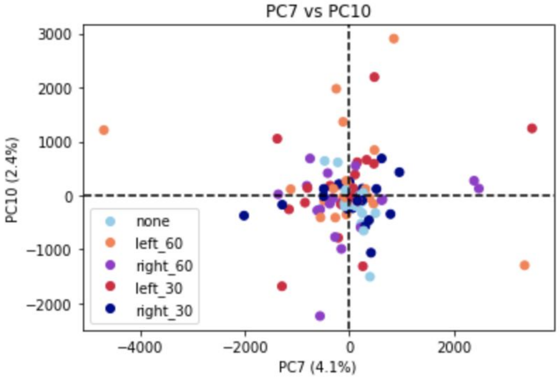


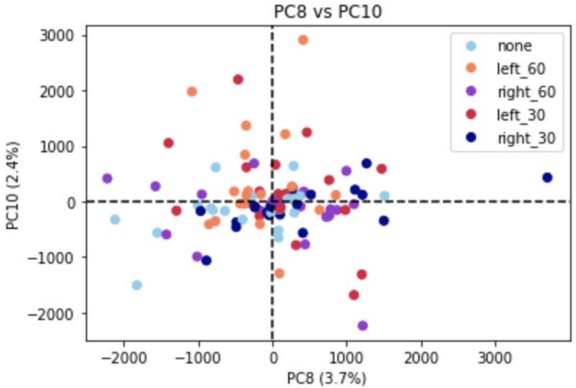
Additional Slides - PCA 310 Channels 5 Conditions



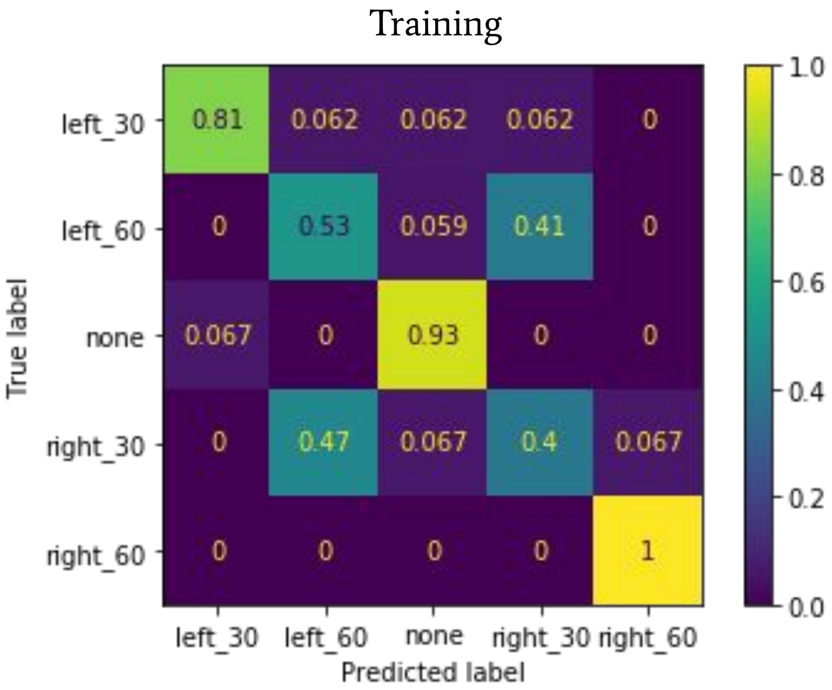


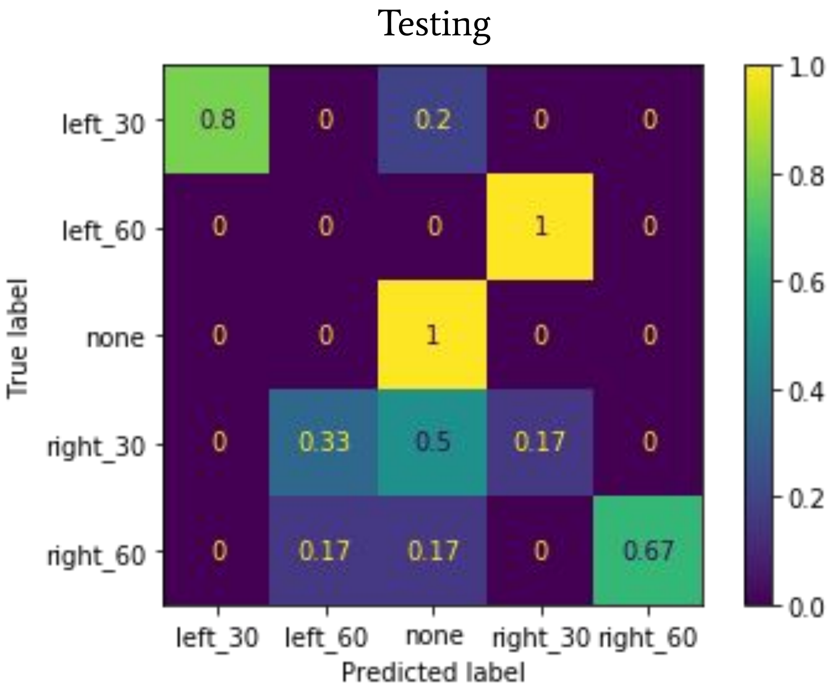




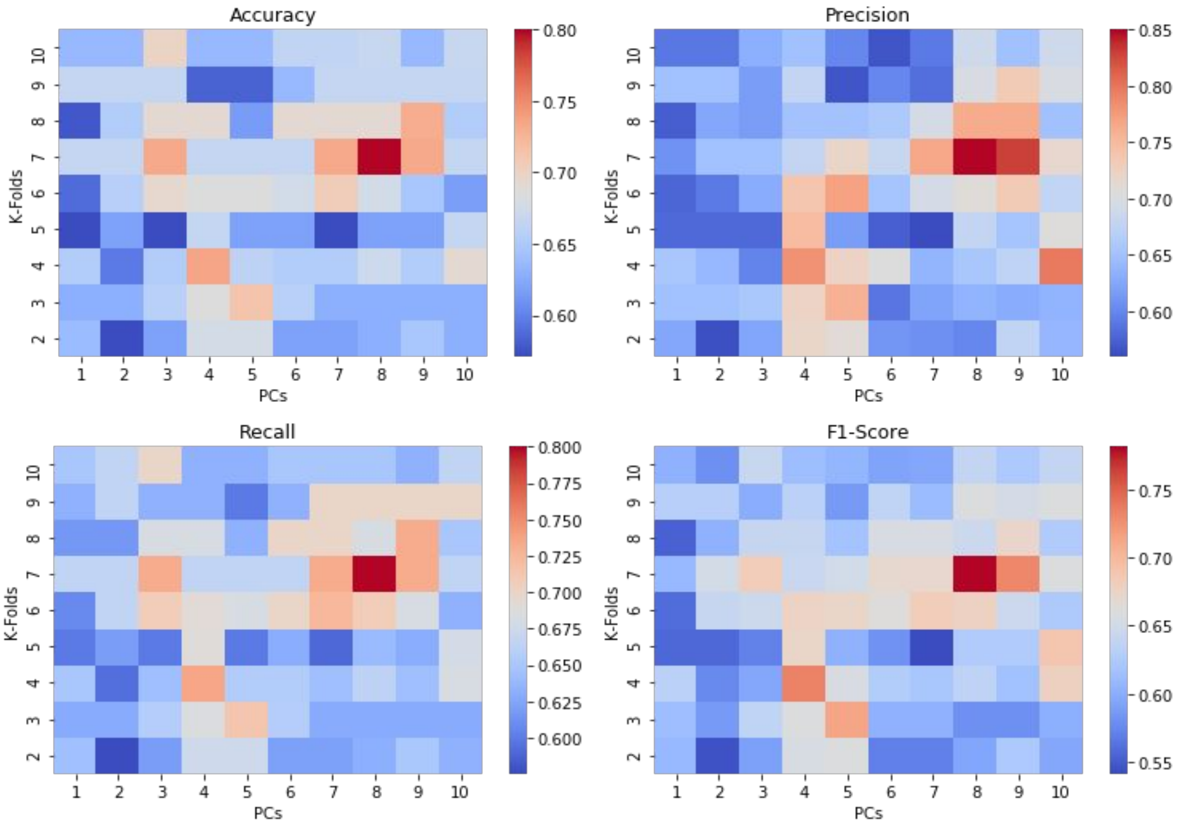


Confusion matrix 310 channels, 5 groups





Heatmaps - 310 Channels, 5 groups



Example bad channels