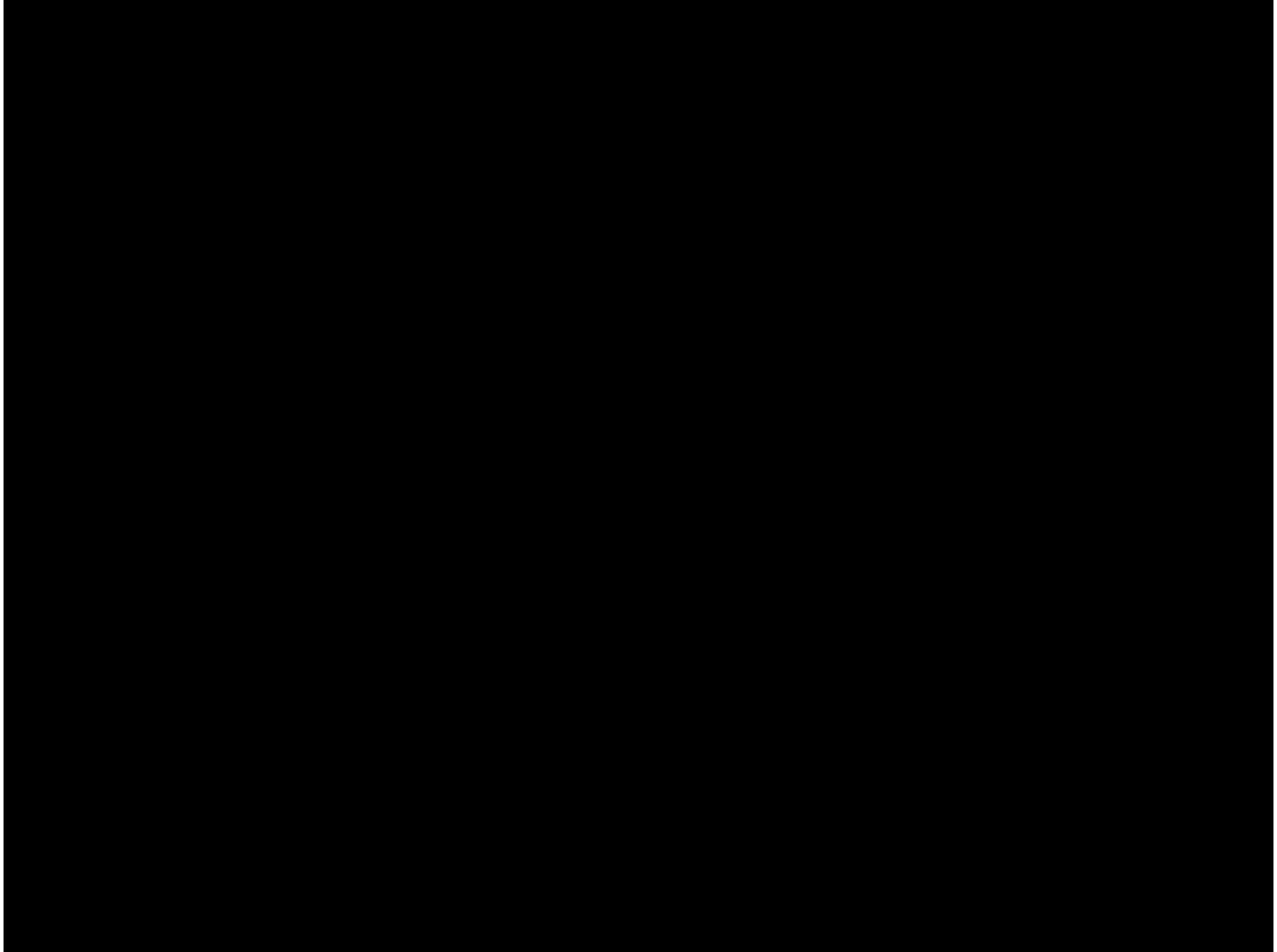
Applications of PCA and Clustering to calcium imaging recordings



What is a video abstractly

Matrix

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

Set of Matrices

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

t=3

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

t=2

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

t=1

Matrices to Vectors

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)

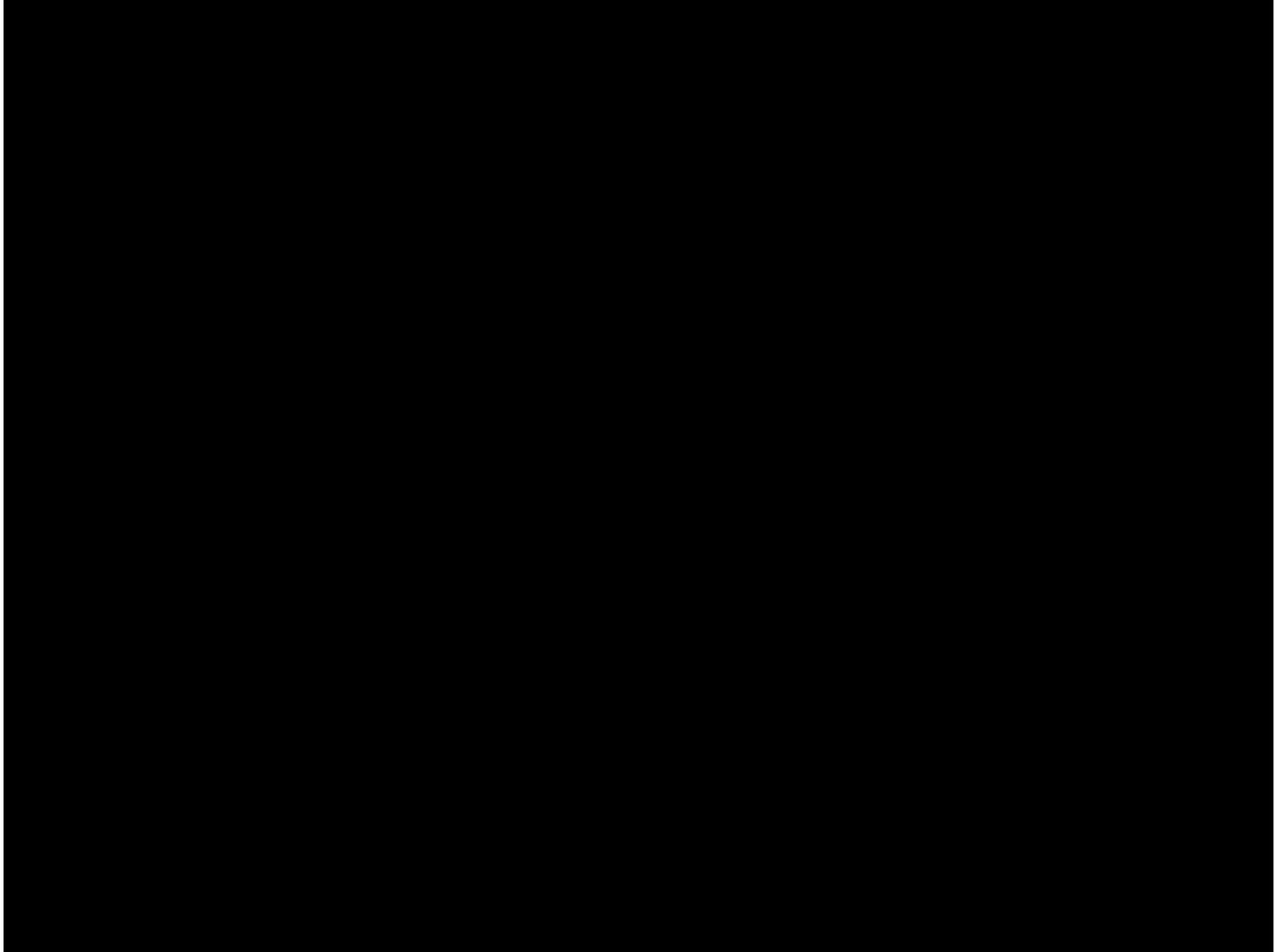
$t=t_n$

$t=t_n$

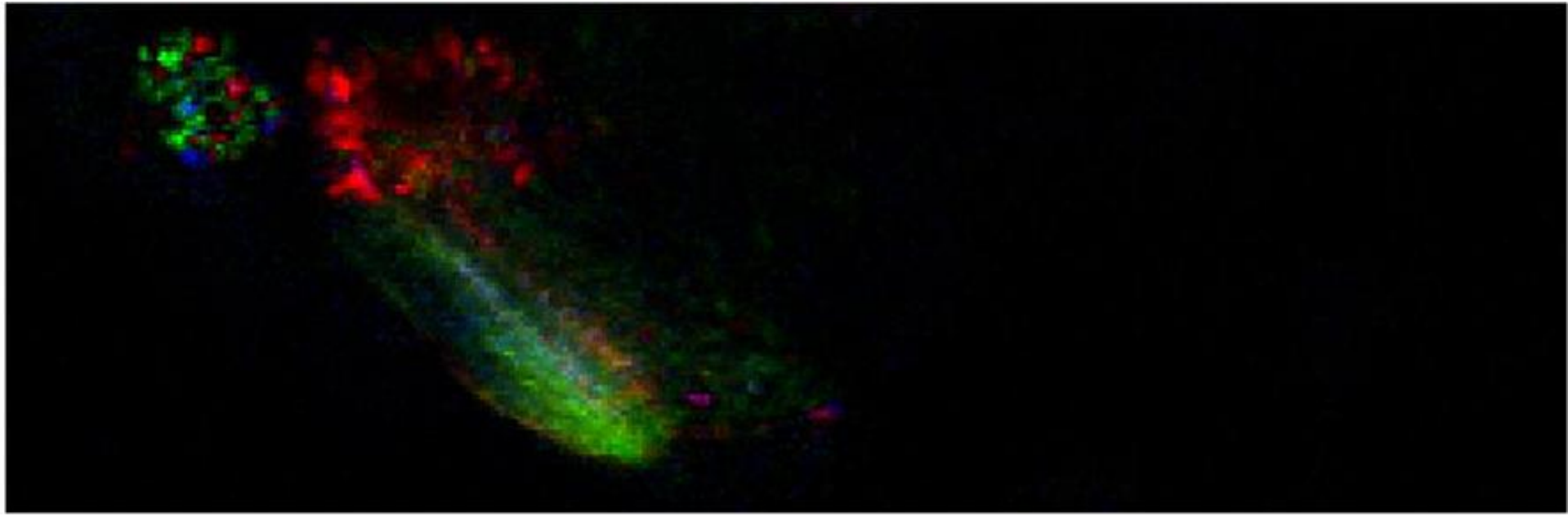
Pixel →

V = Time ↓

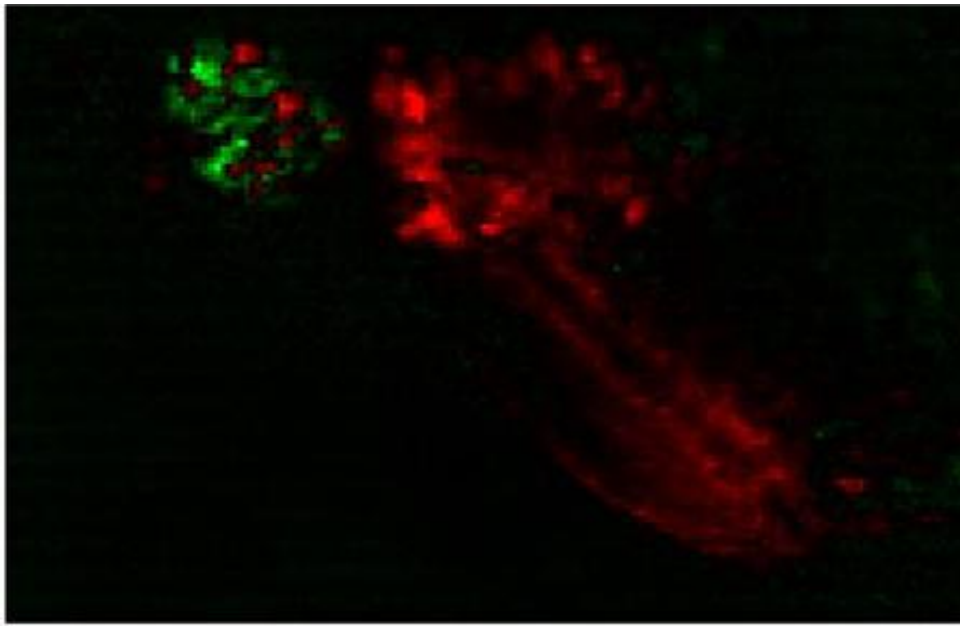
t=1	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
t=2	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
t=3	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
t=4	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
t=5	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
t=6	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
t=...					…….				

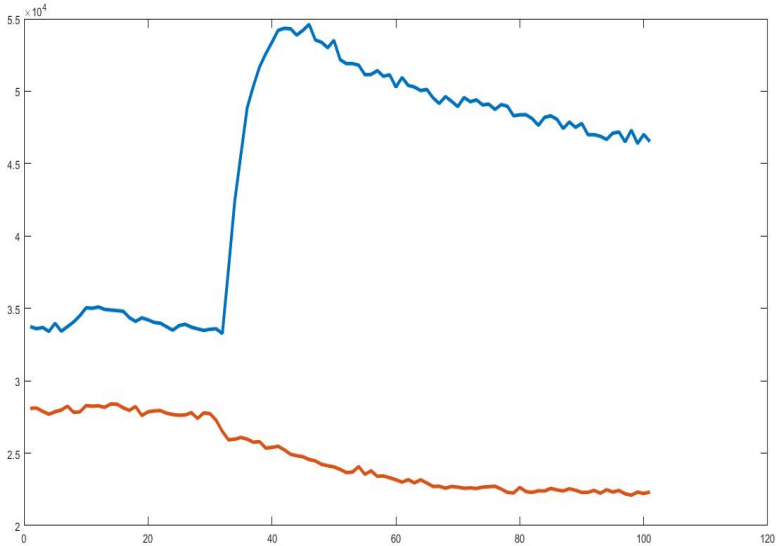


PCA(V) ---> First 3 Principal Components

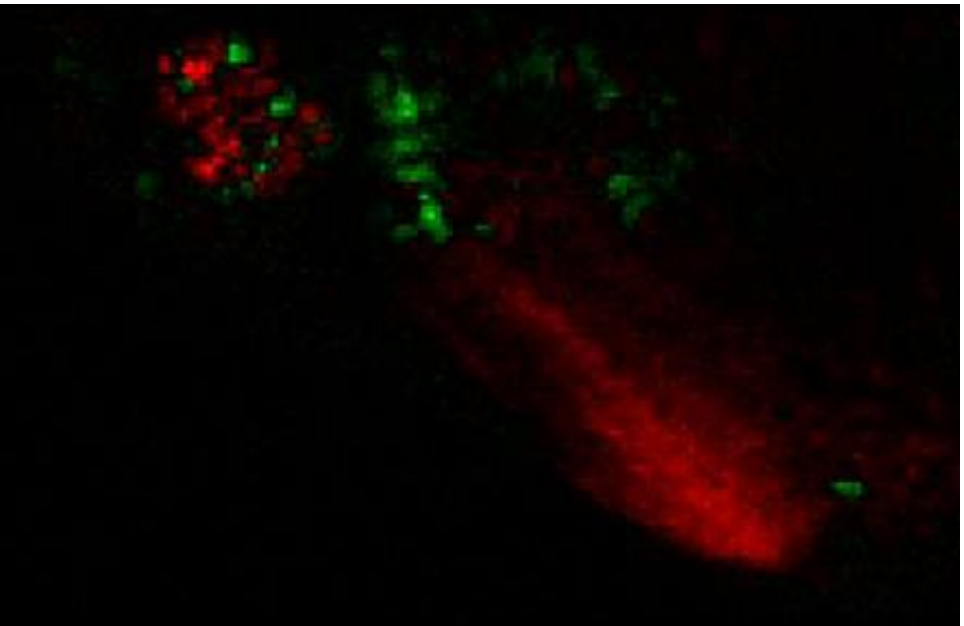


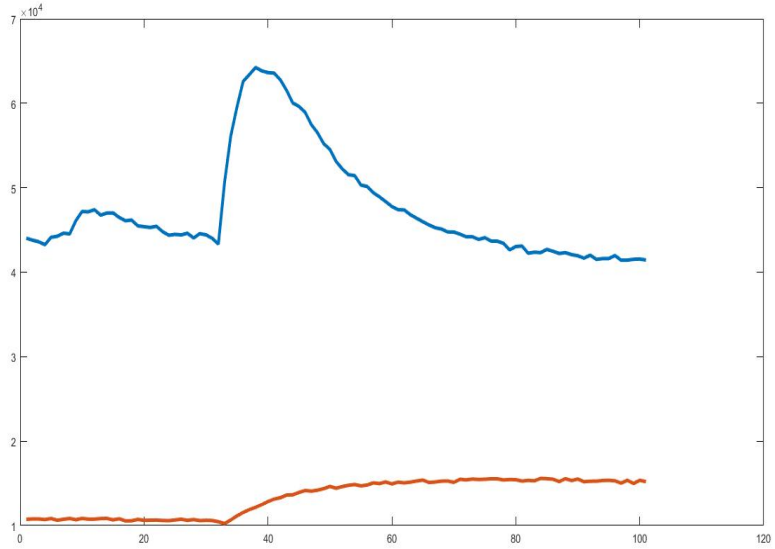
PC1
Red = pixels with positive loadings
Green = pixels with negative loadings



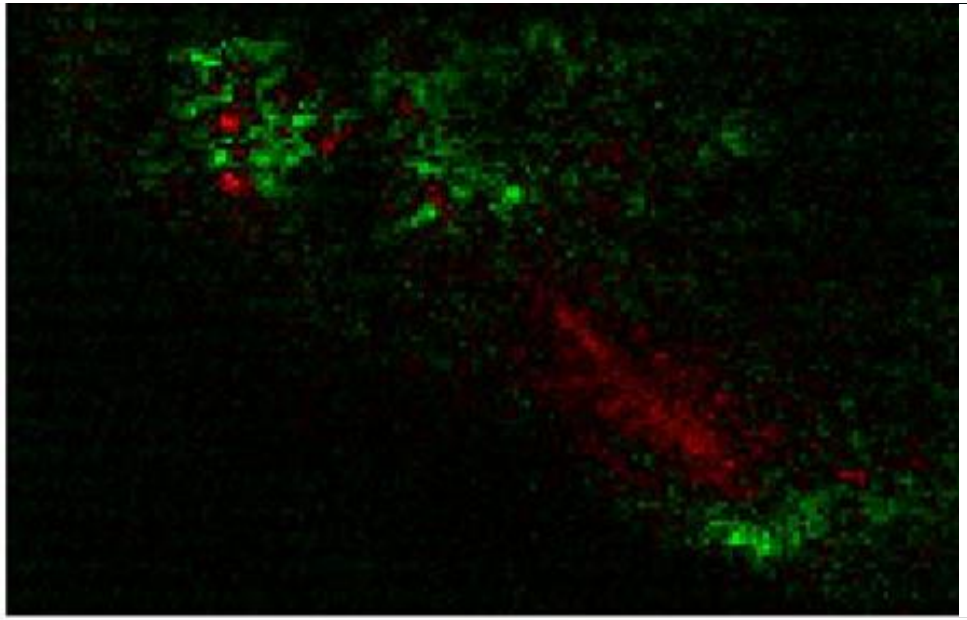


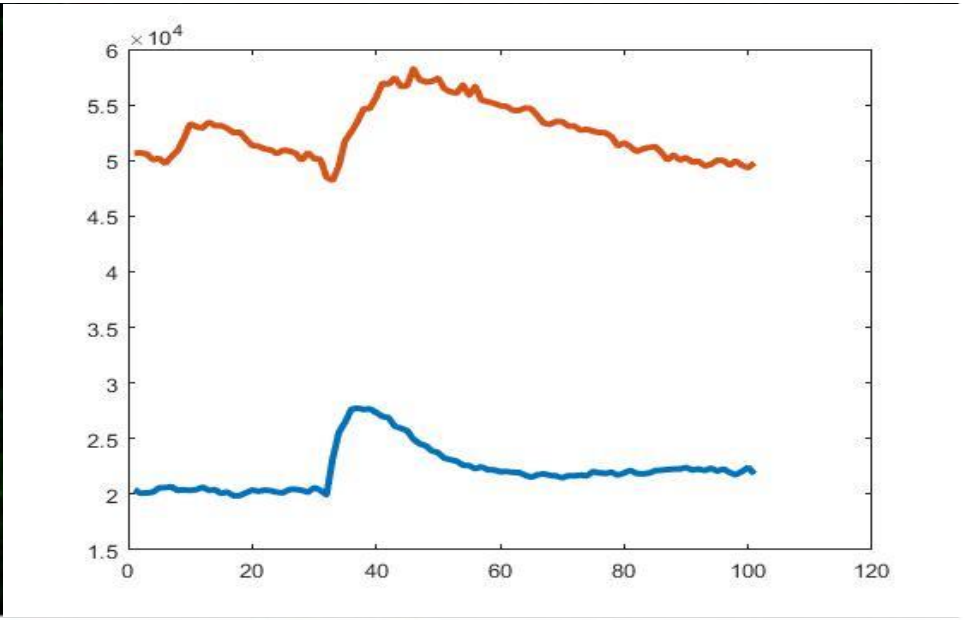
PC2
Red = pixels with positive loadings
Green = pixels with negative loadings





PC3
Red = pixels with positive loadings
Green = pixels with negative loadings





Issues with PCA approach

- Not all pixels are necessarily related, so loadings on every pixel aren’t really meaningful

- How to interpret negative loadings?

Clustering vs PCA

PCA loadings are Eigenvalues of Covariance matrix

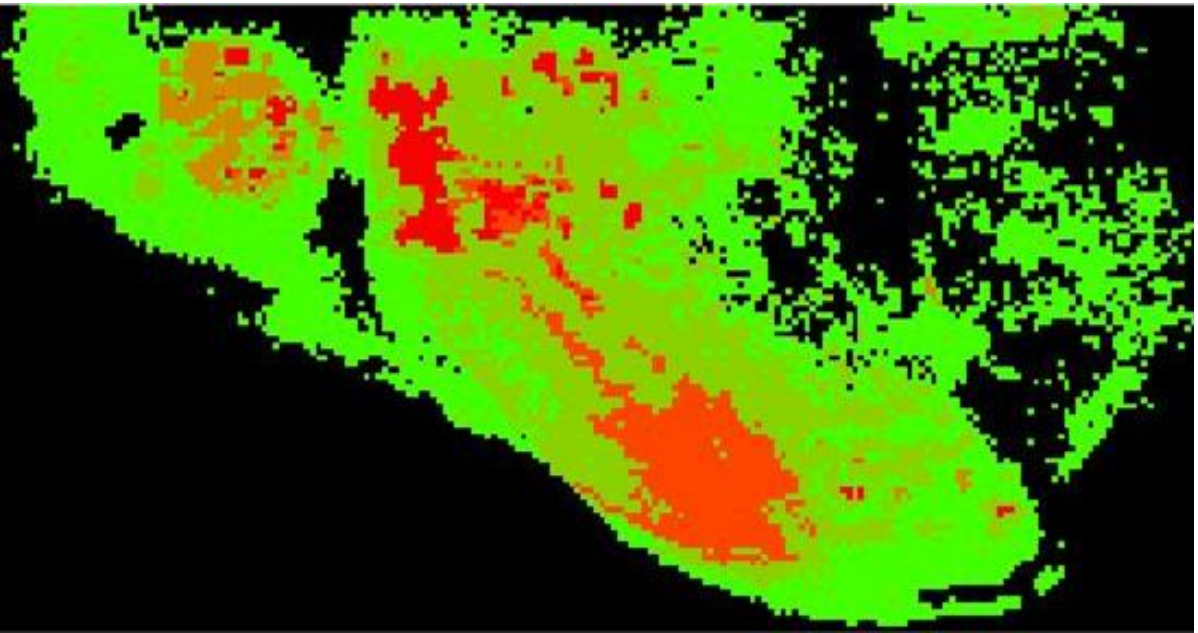
Hierarchical Clustering groups pixels together based on distances between points

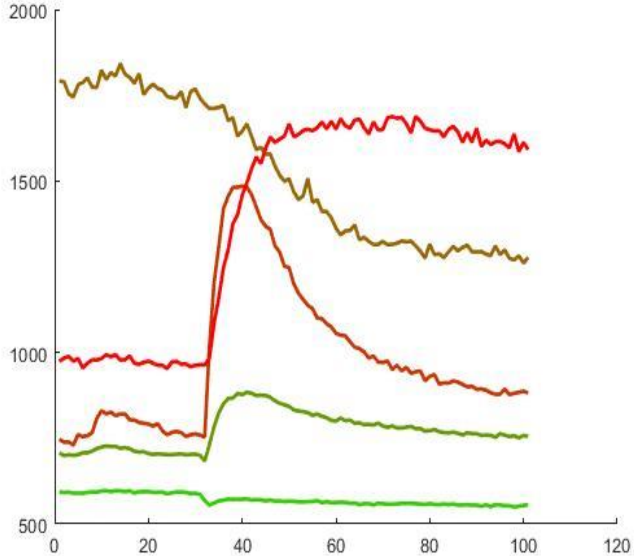
Clustering can be thought of as loadings with binary weights - “No half pixels, no negative pixels”

Clustering might be better for segmentation, since a pixel wouldn’t be .3 in one cell, and .7 in another cell, it’s all or nothing

Why not try clustering covariance matrix?

Clusters of covarying pixels





What about clustering in the row space?

A row in the V matrix corresponds to a single frame in the video

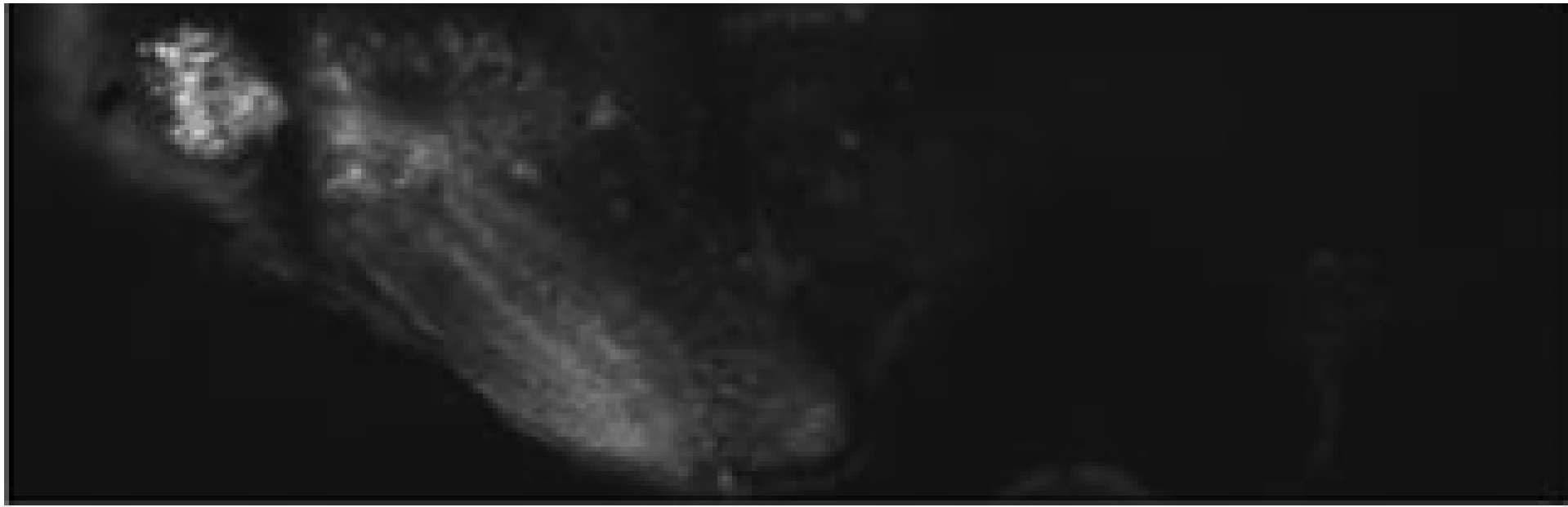
Because there are a finite number of pixels in the image, and a finite number of possible pixel values, there are a finite number of possible frames that can exist

I would argue that Frame → Brain state

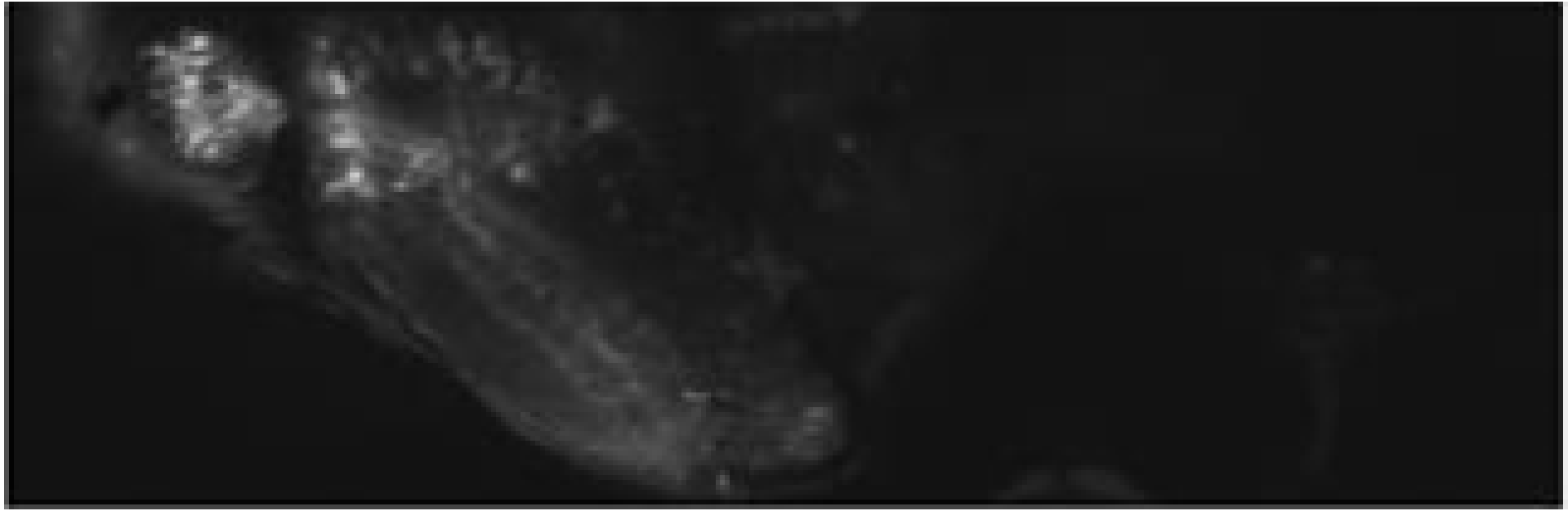
Can think of points in this space as a code that defines a unique brain state

Thought process would then be a parameterized line through this space

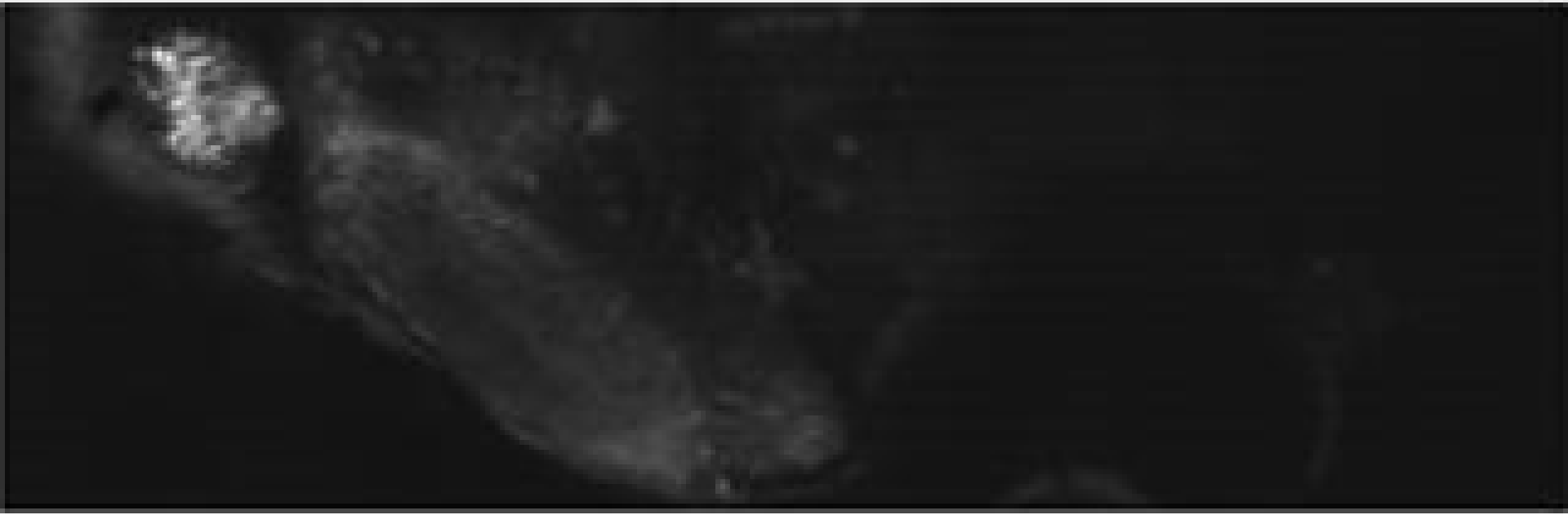
Cluster 1

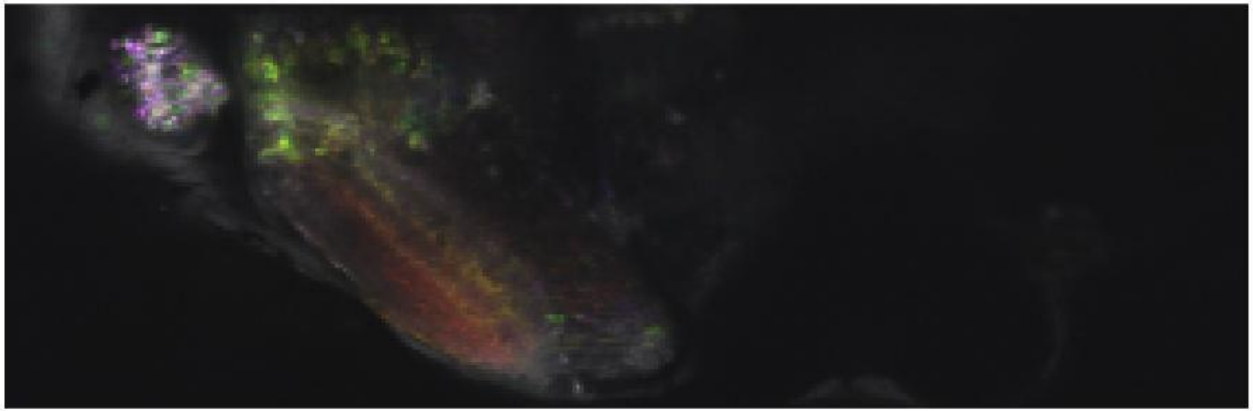


Cluster 2

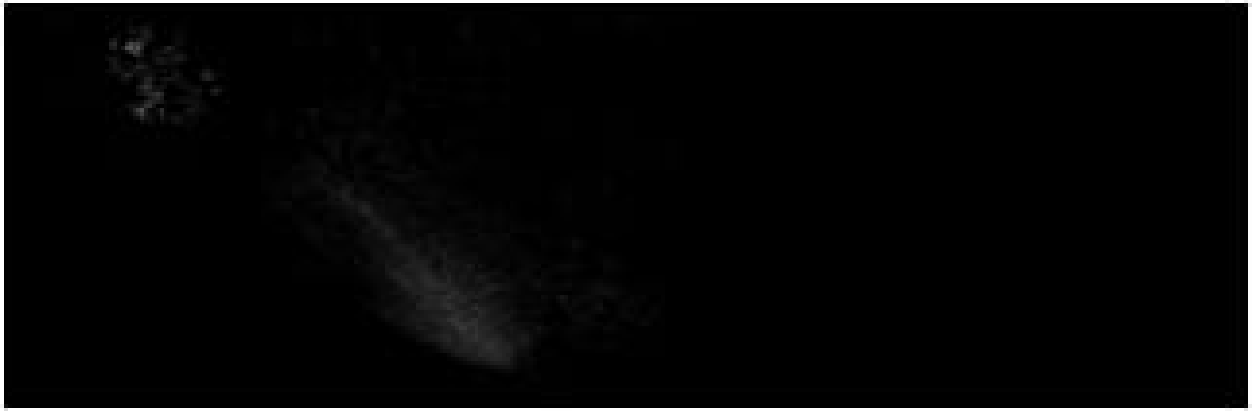


Cluster 3

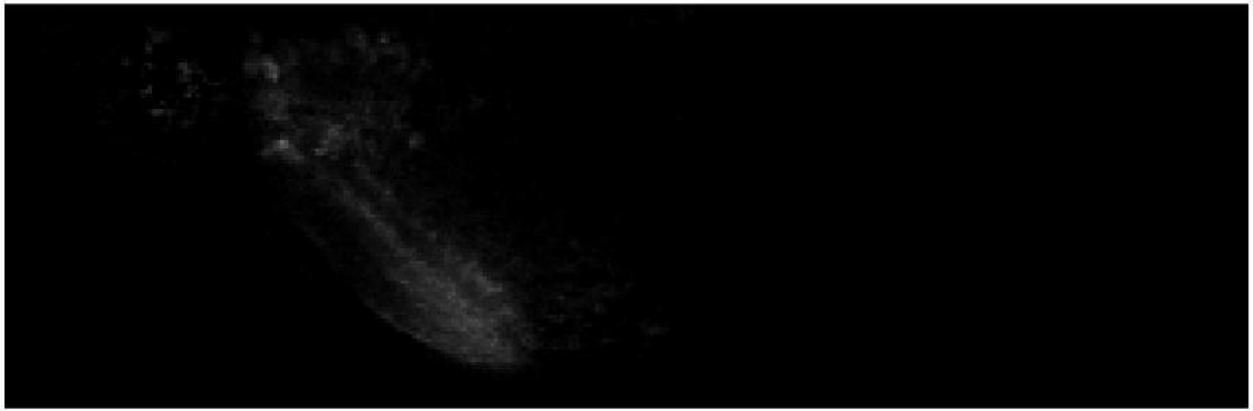




Difference between Clusters 1 and 2



Difference between Clusters 1 and 3



Difference between Clusters 2 and 3