The Alan Turing Institute

# Data Study Group Final Report: UK Dementia Research Institute and DEMON Network

## 6 – 24 Sep 2021

Modelling amyloid beta plaque formation in Alzheimer's disease

https://doi.org/10.5281/zenodo.6798982

# Contents

# 1 Executive Summary

## 1.1 Challenge Overview

Alzheimer's Disease (AD) is a neurodegenerative disorder contributing to $50 - 75\%$ of all dementia cases. Amyloid plaques are accumulations of beta-amyloid proteins that aggregate between the nerve cells (neurons) in the brain of patients with AD and hence, their existence are salient pathological indicators for the disease. Previous research have found that an imbalance between production and clearance of the Amyloid-beta (A$\beta$) and related A$\beta$ peptides is a very early, often initiating factor in AD [19]. Interestingly, a number of different plaque morphologies have been reported to correlate with different clinical features of AD. However, the relationship between these plaque features to the neurodegenerative process remains a central question in AD research.

Despite advancements in transcriptomics methods including high throughput single cell sequencing, for which analysis infrastructure is relatively well established, conventional transcriptomics methods commonly omit the spatial structure of gene expression within an underlying tissue. Recent technological developments in spatial transcriptomics (ST) allows the measurement of the expression of all genes in a tissue and retains spatial information with $100$ micron resolution. This technical advancement has opened new opportunities to investigate the relationship between the amyloid morphological pattern and changes in topological gene expression, such as the genomic responses to the pathological features, the cell types and sub-types contributing to those responses, its dependency to the neighbouring tissue, etc. However, this new form of transcriptomics data has lead to unprecedented data analysis challenges, requiring the combination of two disparate data types: the expression level of several thousands of genes as well as their spatial information including general histology and pathological staining.

In this challenge, we have attempted to understand the relationship between amyloid plaque image and spatial transcriptomics patterns in Alzheimer's disease mouse model using a range of machine learning methods. We have approached by first characterisation and extraction of the key features in spatial transcriptomic and A$\beta$ plaque stained images

separately, followed by a brief exploration of the relationship between the two, and finally the comparison of machine learning models that predict plaque information from gene expression information.

## 1.2 Data Overview

The dataset consists of mouse brain image data accompanied with the spatially corresponding transcriptomics information [3]. It consists of three adjacent coronal slices from $10$ Alzheimer's Disease and $10$ Control mouse brains with $3$, $6$, $12$, and $18$ months of age. The middle slice contains the Spatial Transcriptomics (ST) information, while two outer adjacent slices contain the immunostaining information. Each of the $20$ coronal slices contains more than $500$ transcriptomics profiles of individual tissue domains (TDs). Each TD is annotated with spatial, pathological, and cellular information. The three slices are also aligned to each other in order to annotate each TD with A$\beta$ load, reactive astrocytes, presence of neurons, and nuclei.

## 1.3 Main Objectives

In order to address our proposed challenge of modelling A$\beta$ plaques in AD, we have defined the following $3$ objectives:

1. **Extract and explore the amyloid beta plaques:** In order to explore the various morphologies of A$\beta$ plaques, the key data preprocessing step is to extract them from the A$\beta$ staining brain slide images. This facilitates the application of machine learning methods for the embedding and/or clustering of plaque images, allowing the extraction of potential biologically relevant morphological features of plaques that are not currently captured by manual classification approaches.
2. **Explore and reduce dimensions of transcriptomic data:** Raw transcriptomic data is very high dimensional, with overlapping genes across different cell types and functions. We set out to compare methods of gene classification and dimensionality reduction with methods previously utilised on the data. This has provided further

information about the nature of gene expression pattern, as well as a lower dimensional dataset for relating to amyloid plaque images.

3. **Explore the relationship between genetic data and plaque data:** The final objective is to apply the results of Objectives 1 and 2 to explore the relationship between gene expression and amyloid plaques.

## 1.4 Approach

Image regions containing individual plaques are extracted from A$\beta$ staining images using automated image analysis techniques, including blurring, thresholding and segmentation of the Region of Interest (ROI) surrounding the A$\beta$ plaques. Using the extracted A$\beta$ plaque images, we have explored a number of semi/unsupervised machine learning methods for the extraction of key morphological features from A$\beta$ plaques. These include principle component analysis (PCA) to obtain lower-dimensional feature representation, pretrained VGG-16 convolutional neural network to extract key visual features from the plaque images, and variational autoencoder (VAE) to generate a non-linear lower-dimensional representation of plaque images. To extract lower-dimensional representation of genetic variations across spatial transcriptomics spots, dimensionality reduction techniques, namely, the PCA, $t$-distributed stochastic neighbour embedding ($t$-SNE), uniform manifold approximation and projection (UMAP), and autoencoder (AE), are tested and compared to the previously established methodology, weighted gene co-expression network analysis (WGCNA). A number of clustering and regression methodologies are then evaluated for their performance on the prediction of plaque information.

## 1.5 Main Conclusions

We have established a working procedure for plaque ROI extraction from the A$\beta$ immunofluorescent brain images, and produced promising lower dimensional representation of plaque morphologies by PCA and VAE that appears to outperform the previously established plaque index in encapsulating biologically relevant information. A detailed comparison between various dimensionality reduction methodologies on the gene

expression data has been executed, showing that PCA performs favourably compared to the $t$-SNE, UMAP, AE, and the previous WGCNA method. Under default model hyperparameters, support vector classification (SVC) on the PCA-represented gene expression best predicts plaque presence, while ridge regression on WGCNA and $k$-nearest neighbours on PCA-embedded gene expression provide best prediction on plaque index.

We have explored a number of methodologies to extract key features from plaque morphologies and transcriptomics data. This allows future efforts to employ more explainable machine learning models, e.g. regression, tree-based models, VAE, etc., to learn the relationship between the extracted plaque morphologies and transcriptomics information.

# 2 Data

The dataset consists of mouse brain image data accompanied with the spatially corresponding transcriptomics information [3]. It consists of three adjacent coronal slices from $10$ Alzheimer's Disease (AppNL-G-F) and $10$ Control (C57BL/6) mouse brains with $3$, $6$, $12$, and $18$ months of age. The middle slice contains the Spatial Transcriptomics (ST) information, while two outer adjacent slices contain the immunostaining information. Each of the $20$ coronal (middle) slices contains more than $500$ transcriptomics profiles of individual tissue domains (TDs), adding up to $10,327$ transcriptomics profiles over $20$ coronal sections. The diameter of an TD is $100$ micron and the thickness of a slice is $10$ micron; hence, it is reasonable to assume that cells in the central slice are exposed to amyloid plaques detected in the adjacent slices. Each TD is annotated with spatial, pathological, and cellular information. Each coronal section is aligned with $14$ anatomical brain regions defined by the Allen Mouse Brain Atlas [9], and each TD is assigned to one of them. The three slices are also aligned to each other in order to annotate each TD with amyloid beta load (6E10 staining), reactive astrocytes (GFAP), presence of neurons (NeuN), and nuclei (DAPI).

## 2.1 Image Data

Three adjacent coronal slices ($\sim 20k \times 20k$ pixels) from the middle section of the brain were collected for each mouse. Multi-modal staining was applied on each mouse slide generating nine images as follows:

- H&E staining on the central (middle) slide showing global histology.
- A$\beta$ staining for amyloid beta plaques on two adjacent slides; for the Alzheimer's disease mice only.
- GFAP staining for astrocytes on two adjacent slides .
- NeuN staining for neurons on two adjacent slides .
- DAPI staining for nuclei on two adjacent slides.

All three coronal slices for each mouse are manually aligned to the same pixel space. Figure 1 presents an example of the multi-modal staining

information on the same coronal slice space for an AD mouse.

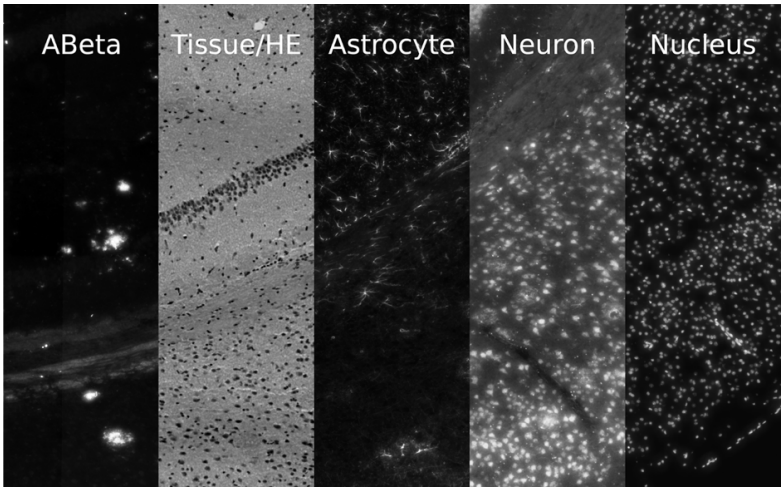



Figure 1: Coronal slice of an AD mouse brain with stainings for amyloid beta plaques, global histology (H&E), astrocytes, neurons, and nuclei.

## 2.2 Localised Gene Expression Data

Each central coronal slide contains approximately $500$ (ranging from $456$ to $560$) spatial transcriptomic (ST) spots spread evenly at a centre-to-centre distance of $200$ µm over its surface. From each $100$ µm diameter spot, RNA has been isolated from the underlying tissue and sequenced, producing an averaged count of the number of individual RNA (gene expression) underlying each spot for $> 15k$ genes. A differential gene expression matrix of spots ($\sim 500$) by gene ($> 15k$) of log-transformed count of the gene expression is then constructed. Figure 2 shows the global histology on coronal slice for an AD mouse brain along with the locations of ST spots and the expression level of one gene (Apoe).

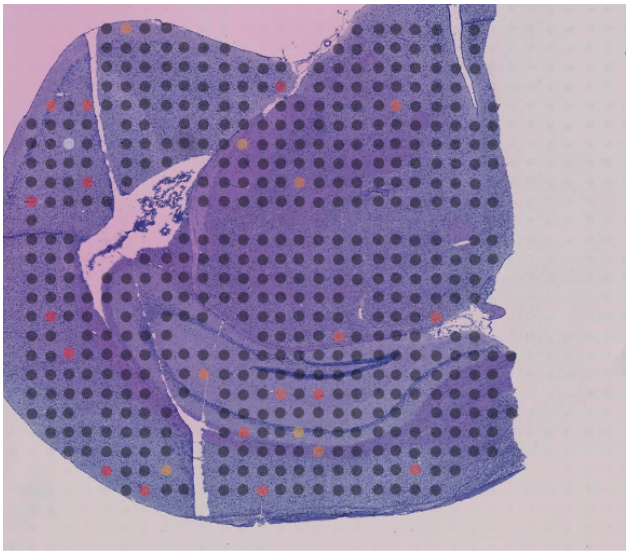


Figure 2: Example of H&E staining on brain coronal slice showing underlying anatomy of an AD mouse. It has been overlaid with the position of spatial transcriptomic spots and the expression level of Apoe gene.

# 3 Objectives

The primary objectives of this project are three-fold:

1. Extraction and Exploration of the morphology of Amyloid-$\beta$ plaques.
2. Analysis and Dimensionality reduction of spatial transcriptomic data.
3. Integration of the two approaches for predicting plaque information from ST data.

## 3.1 Amyloid-$\beta$ Plaque Extraction and Classification

Since the Amyloid-$\beta$ plaques of varying shapes and sizes are spread throughout the coronal slides of the AD mouse brain, in order to match them with the spatial transcriptomics (ST) data, our first aim in this project is to automatically identify and extract them using image analysis techniques.

The exploration of plaque morphology can be sectioned into two categories, the first being the generation of a description for plaques and the second being plaque classification.

Plaque descriptors could include more baseline information about plaques such as location, number of plaques in a given region of interest, and plaque visual intensity. This would provide initial information about plaques, which can be used to correlate with transcriptomic data.

Plaque classification is a more complex problem considering there is no metadata available for the A$\beta$ staining, and there is no existing label for the visual data other than the types of staining. Without labels for these plaques, there is no dataset that can directly be interpreted, meaning that the image data would have to be analysed using unsupervised methods. To facilitate the unsupervised classification procedure, a data subset consisting only of the plaques region of interest (ROI) would have to be extracted from the brain slide image containing hundreds of plaques. This preprocessing step is achieved using a combination of computer vision methods, such as thresholding, blurring, clustering and contouring. For certain analyses (e.g. Singular Value Decomposition), additional plaque processing may be required, such as making the plaque images of

uniform size. One way this can be achieved is by padding images with the average of the background, using that border as a buffer.

## 3.2 Spatial Transcriptomic Data Analysis and Dimensionality Reduction

The current data set contains spatial transcriptomic areas overlaid on the slides. These ST spots contain information regarding genetic expression in the outlined region of $100$ micron diameter. Exploration of the principal component analysis (PCA), manifold learning such as $t$-distributed stochastic neighbour embedding ($t$-SNE) and uniform manifold approximation and projection (UMAP), and the autoencoders, has been undertaken to compare their efficacy in reducing the dimensions of this data. The intention is to compare various unsupervised approaches for dimensionality reduction with prevailing methods used. The objective also creates lower-dimensional feature vectors for our subsequent objective.

## 3.3 Integration of Plaque Image and Spatial Transcriptomics Data

The final objective is to build a model predicting the plaques from the localised gene expression data. We have approached this part of the challenge by first exploring the relationship between the extracted features of plaque morphologies and gene expression data using cross correlation on individual pairs of features. This is followed by the examination on a range of conventional machine learning methodologies for the prediction of plaque presence and plaque index using lower-dimensional gene expression information.

# 4 Plaque Image Extraction

Given the Amyloid-$\beta$ plaque staining images, two key image datasets need to be extracted for our subsequent processing. The first dataset is the extracted Region of Interest (ROI) at the location of the Spatial Transcriptomic (ST) spot data. These spots are the regions on the coronal slides of AD mouse brain where gene expression information is available and hence, they provide direct correspondence between the extracted plaque information and the gene expression data. Our second key dataset is the extracted regions of Amyloid-$\beta$ plaques from the Amyloid-$\beta$ load (6E10 staining) images. This new dataset will be useful for exploring natural groupings in plaque structures and for identifying new plaque sub-types using unsupervised learning approaches.

## 4.1 Extraction of Plaque Images at ST Spot Regions

The metadata available (in the form of a TSV) outlines the location of each spatial transcriptomics spot centre for each given slide. Given the `SampleID` and coordinates of the ST spots, an ROI of fixed size around the spot area can be automatically extracted. Automated image analysis methods such as thresholding, segmentation, and clustering can then be employed to provide information about plaques (if present) within each ST spot. The library/packages used for thresholding and segmentation are incorporated from the OpenCV/CV2 [24] and the clustering methods from sklearn [16]. Figure 3 presents an example of the automated ST spot extraction from the N05_C2 AD mouse brain. We have used the fixed square window of size $200$ pixels for the extraction of each ST spot. We note that although all spots will contain corresponding ST data, not all spots will contain the plaques.

## 4.2 Extraction of Individual Plaque Images

For our automated extraction of Amyloid-$\beta$ plaques from 6E10 staining images, we have explored the following classical image segmentation techniques.

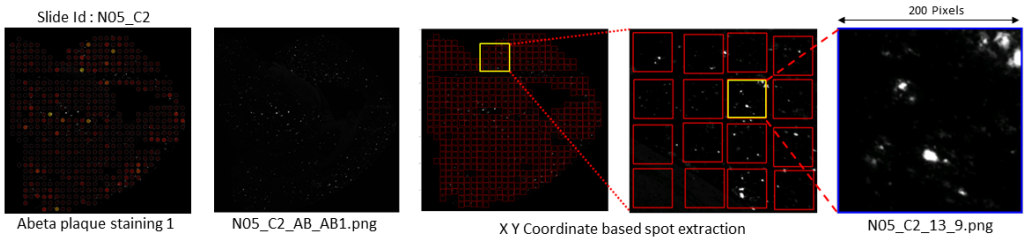



Figure 3: ST spot extraction from the Amyloid-$\beta$ load slide of N05_C2 AD mouse brain.

### 4.2.1 Thresholding

Thresholding is one of the simplest approaches for image segmentation. The procedure usually focuses on partitioning an image feature space (often intensity) into a finite number of classes based on the estimated (or pre-defined) thresholds. In binary (two-class) thresholding, the image pixels are replaced with a black pixel if the image feature (intensity value) $I_{i,j}$ at coordinate $(i, j)$ is less than some fixed threshold T (*i.e.*, $I_{i,j} < T$), or a white pixel if the image intensity is greater than that threshold.

One of the most popular global thresholding techniques is the Otsu thresholding [15]. In its simplest form, the algorithm assumes the foreground pixels belong to one peak in the histogram and the background belong to another peak. In essence, it finds the grayscale value that maximally separates between the two peaks. Owing to the specificity of the A$\beta$ staining, most visible pixels on the A$\beta$ coronal slide in current dataset are plaque signals.

However, in other circumstances, for example, images with considerable amount of noise, a more sophisticated approach than global thresholding is required. Adaptive thresholding methods tend to handle data with variation in brightness much better than a regular global threshold. While the global threshold is applied to the entire image, adaptive thresholding considers groupings of pixels in neighbouring regions and computes an individual threshold value for each of these regions, thus allowing to take into account spatial variation in brightness. In our current study, two types of thresholding have been explored, namely, the adaptive mean thresholding and adaptive Gaussian thresholding.

Adaptive mean thresholding calculates the threshold value based on the mean of several neighbouring areas, while adaptive Gaussian thresholding uses a weighted sum of these adjacent areas where the weights are generated using a Gaussian window [26]. In order to demonstrate the performance of different thresholding methods, we have first selected a cropped region of an Amyloid-$\beta$ stained image in Figure 4.

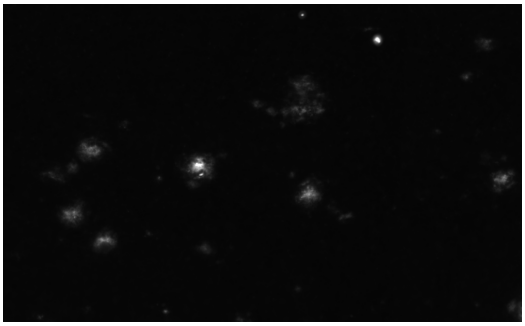


Figure 4: An example of a cropped region of an A$\beta$ plaque stained image.

The qualitative performance analysis of the three thresholding methods, namely, the global thresholding, adaptive mean thresholding, and adaptive Gaussian thresholding, is illustrated in Figure 5.

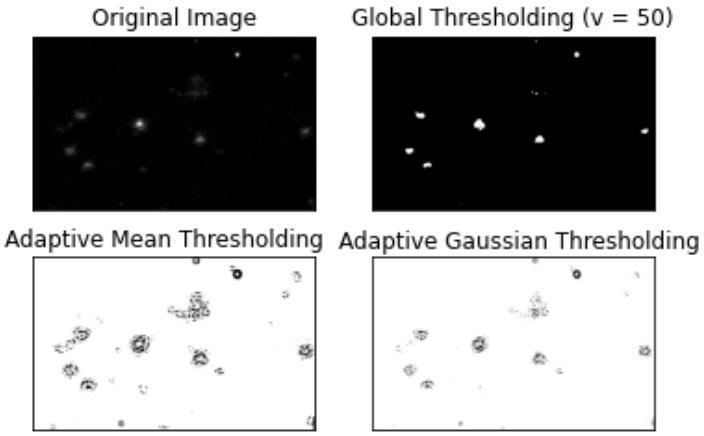



Figure 5: Example of three different thresholding methods, with global thresholding set to 50.

Based on our analysis, global thresholding is sufficiently adequate on our A$\beta$ dataset, due to the high specificity of the A$\beta$ staining. In our current dataset, global threshold of $50$ can adequately capture the A$\beta$ plaques; however, it can be lowered to allow for dimmer structures to be included. Other techniques such as blurring can be considered in future during the preprocessing step to incorporate more diffused plaques structures.

### 4.2.2 Contouring

Contouring is the process of creating an outline around an object. There exists multiple ways of manually creating outlines of a certain shape (through the use of `cv.rectangle()` or in general `cv.drawContours()` in OpenCV) for delineating the plaques in A$\beta$ plaque stained images. However, in our current project we focused only on the automated processing and delineation of the plaque structures based on unsupervised techniques, while the contouring can be useful in future for manually extracting the plaque structures to train supervised (machine learning based) approaches.

### 4.2.3 Plaque Clustering

Clustering is the process of grouping a set of objects depending on certain characteristics, such that the groupings of objects (clusters) are similar in terms of the said characteristics. Clustering has been used throughout this study for different purposes. In this case, it is implemented to measure the number of plaques in an image. As our objective is to find the number of clusters (individual plaques), popular methods like the $k$-means clustering would not be appropriate due to its requirement of number of clusters as an input.

One of the most popular clustering methods that does not require a pre-defined number of clusters is *Density-based spatial clustering of applications with noise* (DBSCAN). It is a non-parametric density-based clustering approach. Given a set of observations in a spatial system, the algorithm aims to automatically group points that are closely packed together. On our case of plaques within A$\beta$ stained images, the groupings of points within the plaque regions are largely concentrating, allowing for DBSCAN to be a good alternative. However, DBSCAN still requires a

minimum cluster size and a distance threshold as user-defined parameters. An alternative method that extends on DBSCAN is the Hierarchical DBSCAN (HDBSCAN) [11], requiring only the minimum cluster size to be specified as an input. The performance of the HDBSCAN clustering method on our example case of A$\beta$ plaque stained image from Figure 4 is presented in an interpretable way in Figure 6. All generated clusters are coloured, with the dimmer colours being less likely to be classified as a cluster (plaque in our case).

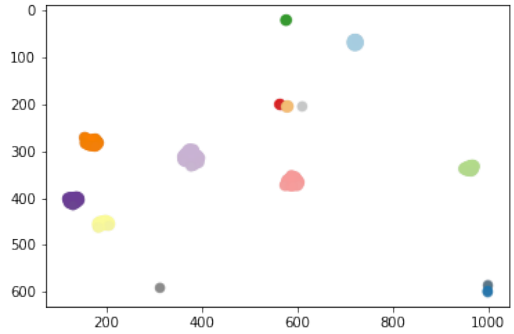



Figure 6: The performance of HDBSCAN clustering on the cropped image.

#### 4.2.4 Plaque Image Segmentation

In order to automatically segment and extract the plaques from A$\beta$ plaque stained images, we have developed an automated approach detailed in Algorithm 1. The script is created partly based on the resources available in [6, 4, 23, 24, 8], and mostly applies the Python library 'OpenCV' [1].

The Algorithm starts with a few preprocessing steps, namely, increasing the image contrast in Line 11, blurring the image with a bilateral filter with Gaussian kernel in Line 12, converting the pictures to grayscale (if necessary) in Line 13, and thresholding to remove pixels with intensity less than the maximum intensity divided by $3$ in Lines 14-15. An image contouring algorithm is then applied on the preprocessed A$\beta$ image in Line 16. As suggested in [6], we use the following settings in 'OpenCV' library:

mode = cv.RETR_TREE, method = cv.CHAIN_APPROX_SIMPLE.

**Algorithm 1** Extracting the plaques from A$\beta$ plaque stained images

```
FilePathArray = []
MetaData = []
for all subdirectory in directory do
    for all FilePathName in subdirectory do
      Append FilePathName to FilePathArray
    end for
end for
for all FilePathName in FilePathArray do▷ This part mostly follows [6]
    Load Picture from FilePathName
    Convert picture from BGR to RGB                          ▷ (if necessary)
    Increase Contrast picture                                ▷ (if necessary)
    Blur Picture     ▷ bilateral filter: d =70, σ_Space = 2000, σ_Colour = 2000
    Convert picture to grayscale                             ▷ (if necessary)
    a = Max(Picture)
    Binary thresholding: eliminate pixels below a/3
    Find Contours
    for all contour in Contours do                  ▷ This part follows [4, 23]
        find rectangle around contour
        if Width(Rectangle) > 3 then
            Find x-coordinate of centroid of rectangle
        else centroid = topLeft x-coordinate(Rectangle)
        end if
        if Height(Rectangle) > 3 then
            Find y-coordinate of centroid of rectangle
        else centroid = topLeft y-coordinate(Rectangle)
        end if
        Append [Cropped plaque Image Name,...
                    ... Centroid coordinates, Rectangle dimension] to...
                    ... MetaData
        Crop PlaqueImage in form of Rectangle
        Write PlaqueImage
    end for
end for
```

Next following [4, 23], we find the smallest rectangle that includes the generated contour in Line 18 and finally, we determine the centroid of the rectangle in Lines 19-26. In order to remove the noise or outlier points as the extracted contours, we check if the height and width of the rectangles are at least $4$ pixels.

An example of the performance of Algorithm 1 for automatically extracting plaques from Figure 4 is presented in Figures 7 and 8. After blurring the picture and applying a threshold to remove noise and background pixels, we can see that the algorithm successfully captures areas that corresponds to plaques in Figure 7. The blurring also allows to capture larger, more diffused areas as the plaque region within one rectangle.

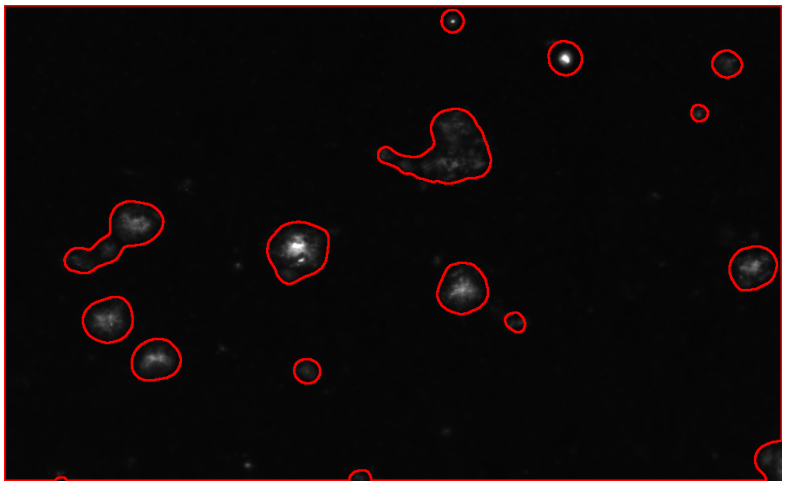


Figure 7: Contouring of the plaque regions: the algorithm successfully identifies most of the clouds of points on the A$\beta$ stained image.

The final performance of generating the rectangular plaque regions as the ROI is presented in Figure 8. The overall plaque extraction approach, as presented in Algorithm 1, takes $2$ minutes $25$ seconds ($\pm 39$ seconds) for each A$\beta$ stained image, in a Windows 10 64-bit OS machine with an Intel(R) Xeon(R) W-2245 CPU at 3.90 GHz and 64 GB RAM.

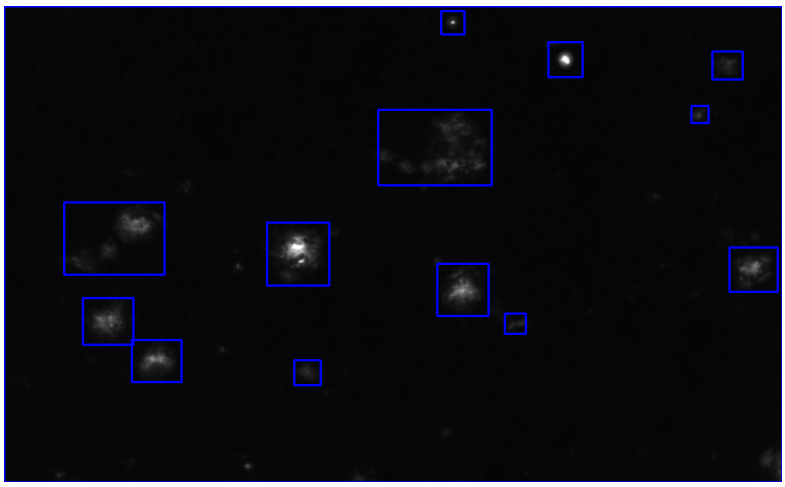


Figure 8: A rectangle is drawn around the contour. The rectangle is used to crop the picture around the cloud of points.

# 5 Meaningful Representation of the Plaque Images

One of the principal aims of this project is the exploration of different subtypes of Amyloid-$\beta$ plaques based on their appearance (morphology). In this regard, a reference metric named *plaque_score* has been used by Chen et al. [3], which calculates the standard deviation of the intensity of all plaques found within the spatial transcriptomic (ST) spot. While this score does capture some information such as the presence of plaques and the intensity variation, some key biologically relevant information can be missed in this metric, including the number and types of plaques in each spot. In order to automatically extract key image features and present meaningful plaque description, we have incorporated the traditional computer vision approaches as well as the state-of-the-art machine leaning based methods including SVD, VGG-16, and VAE.

## 5.1 Plaque Features Extraction

There are a number of interpretable morphological characteristics on the plaque images that can be extracted with relative ease. Some examples are:

1. Plaque size,
2. Plaque distance to the spot,
3. Plaque Shape, e.g. circularity, number of convex structures, holes, smoothness, etc.

Through the use of traditional computer vision approaches, most of these descriptors can be easily computed.

## 5.2 Number of Plaques in a Spot

In addition to the presence of plaque and *plaque_score*, one most biologically relevant information is the number of plaques in the spot region. A straightforward approach to plaque counting is binary thresholding, followed by spatial clustering. Conventional clustering

methods such as $k$-means would require the prior information regarding the expected number of clusters, thus making it inapplicable for identifying varying number of plaques in a spot region. HDBSCAN is a more appropriate alternative of clustering in our case, requiring only the minimum number of clusters as input. In case where no plaque is present in a given spot, the input channel can be filtered out by an initial thresholding, thus preventing the generation of false positive clusters.

## 5.3 Eigen Plaques

Having segmented and extracted the plaques in Section 4, we propose to decompose the pictures to obtain a meaningful representation that may give us some insight on the plaques and, in turn, generate a lower-dimension representation of the dataset that may be more amenable to the subsequent clustering and regression tasks (for example together with the localised gene expression data). We aim to achieve it using the Singular Value Decomposition (SVD) method [25, 13, 17].

As explained in [17], an image $\mathbf{X}$ of size $M \times N$, where $M \geq N$, can be decomposed into:

$$\mathbf{X} = \mathbf{U}_X \times \mathbf{S}_X \times \mathbf{V}_X^T, \tag{1}$$

where:

- $\mathbf{U}_X$ is an $M \times M$ orthogonal matrix composed of the eigen vectors of matrix $\mathbf{X}\mathbf{X}^T$.
- $\mathbf{S}_X$ is an $M \times N$ matrix where the diagonal elements are the singular values of $\mathbf{X}$.
- $\mathbf{V}_X$ is an $N \times N$ orthogonal matrix composed of the eigen vectors of matrix $\mathbf{X}^T\mathbf{X}$.

As explained in [13], we can extend this concept to multiple images. Let us assume we have $P$ images of size $M \times N$. Then the images can be reformatted into $P$ vectors of size $M*N$. We assume that $P \leq (M*N)$ (see [13]). Then we can align the vectors to create matrix $\mathbf{Z}$ of size $(M*N) \times P$. Now, matrix $Z$ can be decomposed using an SVD decomposition:

$$\mathbf{Z} = \mathbf{U}_Z \times \mathbf{S}_Z \times \mathbf{V}_Z^T, \tag{2}$$

where

- $\mathbf{U}_Z$ is an $(M*N)\times(M*N)$ orthogonal matrix composed of the eigen vectors of matrix $\mathbf{Z}\mathbf{Z}^T$. As it is composed of eigen vectors of all the images simultaneously, it can be seen as an orthogonal basis for the images.
- $\mathbf{S}_Z$ is an $(M*N)\times P$ matrix where the diagonal elements are the singular values of $\mathbf{Z}$.
- $\mathbf{V}_Z$ is a $P\times P$ orthogonal matrix composed of the eigen vectors of matrix $\mathbf{Z}^T\mathbf{Z}$.

Turk and Pentland [25] used a similar method for faces and referred to the base images as "eigen faces". By extension, we could call the basis as "eigen plaques".

Before proceeding to the Singular Value Decomposition (SVD), a certain number of preprocessing steps are necessary: eliminating images that are too large or small and images that are too uniform, formatting the images to be exactly the same size, increasing the contrast, and finally reformatting the images into vectors. We describe these steps in the next subsection.

### 5.3.1 Plaque Images Preprocessing

The preprocessing of plaque images consists of two main steps: first, removing the outlier plaque images and second, modifying the images to be of uniform size by padding around the plaque region. The implementation of the two preprocessing steps is described in Algorithm 2.

#### Remove Outlier Images

For eigen plaques to be meaningful, we may want to keep plaque images that are somewhat similar and remove the outliers. Hence, we first aim to remove plaque images that have too little variability or are too large or too small. We perform the following three steps:

1. Calculate the variances of the plaque images in Line 6 of Algorithm 2. A histogram of these variances is presented in Figure 9.

**Algorithm 2** Preprocessing Steps Algorithm

heights = []; widths = []; variances = []
**for** Image in CroppedPlaqueImages **do**
    Read Image
    append Image height to heights
    append Image width to widths
    append variance of Image intensities to variances
**end for**
Minh = 20; Maxh = 100; Minw=20; Maxw = 100; VarMin = 61; ImgArr = []
**for** Image in CroppedPlaqueImages **do**:
    load Image
    **if** (height(Image) $\geq$ Minh) And (height(Image) $\leq$ Maxh) And
        (width(Image) $\geq$ Minw) And (width(Image) $\leq$ Maxw) And
        (variance(Image) $>$ VarMin) **then**
        append Image to ImgArr
    **end if**
**end for**
**for** Image in ImgArr **do**: ▷ This part of the code is based on [8]
    load Image
    ColourMean = Mean(Clr(TopRowImage),Clr(BottomRowImage),
     Clr(LeftColumnImage), Clr(RightColumnImage)) ▷ Clr=colour
    **if** height(Image) is even **then**:
        topadd = $\frac{\text{Maxh-height(Image)}}{2}$; bottomadd = topadd
    **else**
        topadd = $\frac{\text{Maxh-height(Image)}}{2} - 0.5$; bottomadd = topadd+1
    **end if**
    **if** width(Image) is even **then**:
        leftadd = $\frac{\text{Maxw-width(Image)}}{2}$; rightadd = leftadd
    **else**:
        leftadd = $\frac{\text{Maxw-width(Image)}}{2} - 0.5$; rightadd = leftadd+1
    **end if**
    add borders of colour ColourMean of sizes topadd to the top,
    bottomadd to the bottom, leftadd to the left,
    rightadd to the right of Image.
    write Image
**end for**

2. Calculate heights of plaque images in Line 4. A histogram of the heights is presented in Figure 10(a).
3. Calculate widths of plaque images in Line 5. A histogram of the widths is presented in Figure 10(b).

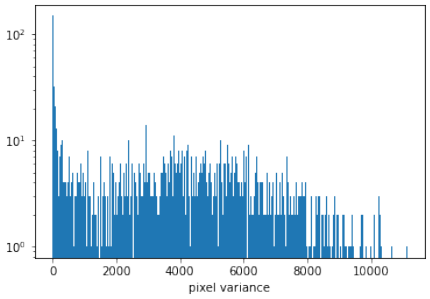



Figure 9: Histogram of the variances of pixel intensities in plaque images.

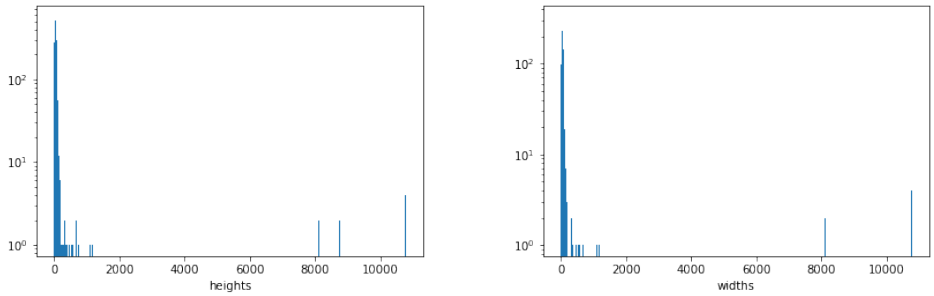



Figure 10: Histogram of (a) heights and (b) widths of the plaque images.

We can use these histograms during our initial preprocessing to remove outlier images. We calculate the quantiles of the variances, heights, and widths to obtain:

- quantile(Variances, 0.1) = 61.
- quantiles(Heights, (0.1,0.9)) = (16,101).
- quantiles(Widths, (0.1,0.9)) = (16,101).

Consequently, we remove:

- plaque images with pixel variance $< 61$ in Line 13 of Algorithm 2,
- plaque images with heights $< 20$ or heights $> 100$ in Line 11, or
- plaque images with widths $< 20$ or widths $> 100$ in Line 12.

**Obtain Plaque Images of Uniform Size**

The SVD technique for plaque features extraction will be applied to a matrix with columns consisting of the extracted plaque images. Since the extracted rectangular plaque regions are of different sizes, we propose including padding of uniform colour around the cropped plaque images in Lines 21-33 of Algorithm 2. To reduce artifacts generated by introducing padding of an arbitrary colour, we propose to calculate the average intensity of the pixels at the border (topmost and bottom-most rows and leftmost and rightmost columns) of the plaque image in Lines 19-20.

A random sample of the generated plaque images after the two preprocessing steps is presented in Figure 11. The overall preprocessing steps, as presented in Algorithm 2, takes 1 minute 3 seconds in a Windows 10 64-bit OS machine with an Intel(R) Xeon(R) W-2245 CPU at 3.90 GHz and 64 GB RAM.

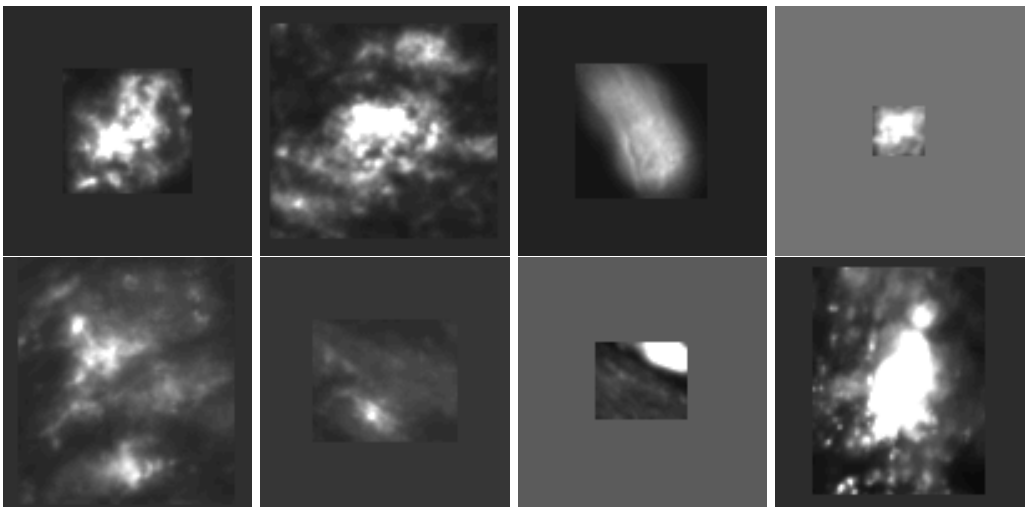


Figure 11: A random sample of generated plaque images after the two preprocessing steps.

### 5.3.2 Singular Value Decomposition (SVD)

In this section, we briefly describe the results of the SVD technique. The implementation of the SVD technique on the plaque images is presented in Algorithm 3.

**Algorithm 3** SVD decomposition

```
1: ImMat = zerosMatrix(
2:       NbRows = height(BorderedImage) × width(BorderedImage),
3:       NbCols = length(BorderedImageList) )
4: for i in 1,···,length(BorderedImageList) do
5:     read Image
6:     ImMat[:,i] = reformat( Image,
7:       NbRows = height(BorderedImage) × width(BorderedImage)
8:       NbCols = 1 )
9: end for
10: U, S, Vh = SVD(ImMat)
```

Our first step consists of reformatting the preprocessed image dataset into a matrix, with columns consisting of the extracted plaque images, as presented in Lines 4-9 of Algorithm 3. We then apply the SVD using NumPy toolkit [14] in Line 10. The overall Algorithm 3 takes $26$ seconds in a Windows 10 64-bit OS machine with an Intel(R) Xeon(R) W-2245 CPU at 3.90 GHz and 64 GB RAM.

As described in [13], $\mathbf{U}$ is the matrix of eigen vectors of the covariance matrix $\mathbf{X}\mathbf{X}^T$. These vectors form a basis for the plaque images. After the application of SVD, we can transform the vectors of matrix $U$ back into the images. These images show us what we can call the "eigen plaques" of the plaque images, i.e. a basis for the plaque images. The first (most important) $20$ eigen plaques are presented in Figure 12.

From Figure 12, we can notice that the most important components of the plaque images are circular shapes. From the fourth basis, we can see some directionality elements, indicating the orientations in the plaques.

As discussed in [17], to assess the importance of each eigen plaque, we take a look at the singular values after the decomposition. The results

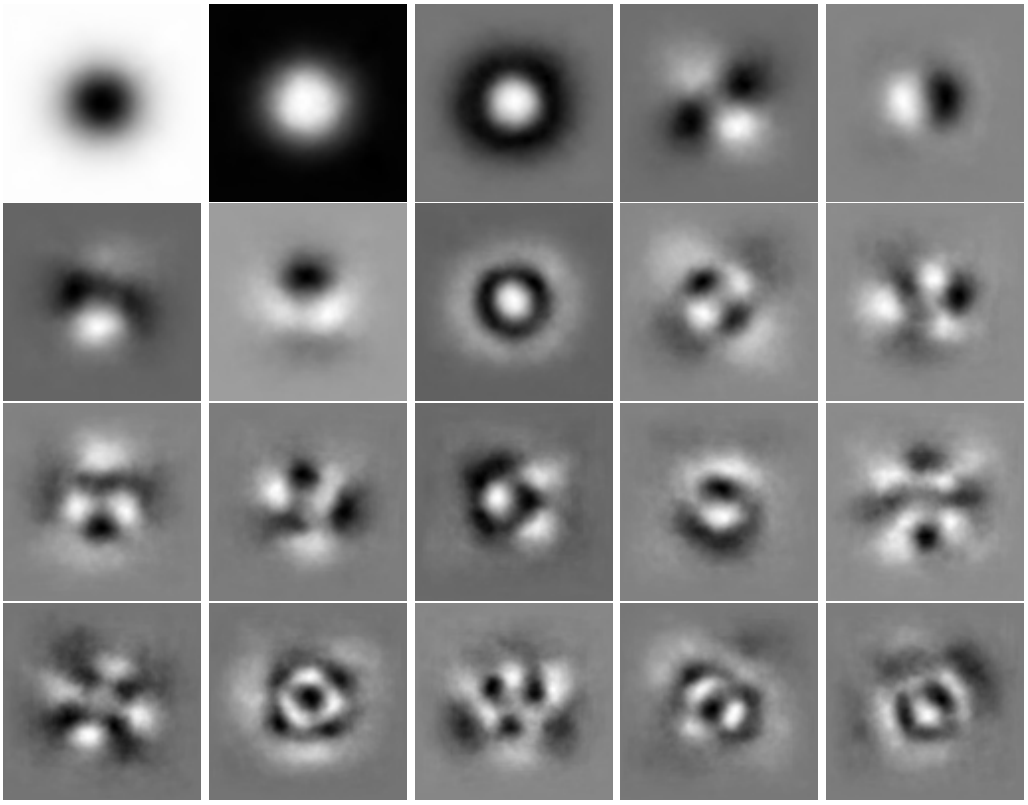


Figure 12: First 20 eigen plaques reformatted as pictures.

are presented as barplots in Figure 13. We notice a rapid decay at the beginning; however, the decay plateaus with a relatively slow decreasing values after the first $7$ basis vectors.

### 5.3.3 Remarks

Further to the analysis in this section, we can make the following remarks: first, we have focused on the left matrix $\mathbf{U}$ to obtain the eigen plaques. As mentioned in [17], the matrix $\mathbf{V}$ containing the eigen vectors of matrix $\mathbf{X}^T\mathbf{X}$ is interesting in its own right. Sadek [17] indicated that the vectors of matrix $\mathbf{V}$ can be seen as directions of critical energy (see also [21] for another example of use of the $\mathbf{V}$ matrix). Second, we have increased the contrast twice in this algorithm. This might be unnecessary and some testing may be required to select the correct contrast. Third, the SVD is usually run

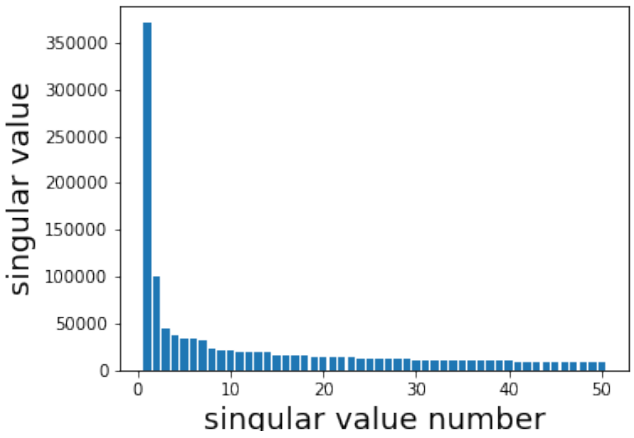



Figure 13: Singular Values of the basis vectors after the decomposition.

after the means have been removed (see [13]), which may change some of the eigen plaques and their interpretations. Next, we have not discussed here the reconstruction of the original plaque images and how many basis vectors are required in order to retain the quality of the original images (see [13, 17]). Finally, the plot of singular values (Figure 13) can be reformatted as in [17, 13]. The singular values can be normalised by dividing with the first one and the values can be presented on a logarithmic-scale on the y-axis.

## 5.4 Extraction and Dimensionality Reduction of Spatial Transcriptomic Spots

A similar approach to that used for the analysis of plaque images is also applied to the extracted spatial transcriptomic spots. The reason is to explore what bases that underpinned the distribution of Amyloid-$\beta$ on the spots, and to generate a lower-dimensional dataset to ease downstream analyses to explore the relationship between image and genetic data.

The similar method is followed as in the plaque data: the $5009$ extracted spots are combined into a matrix that underwent the principle component analysis. The preprocessing step of size correction is not required as all

spots are extracted based on the square window of same size. The first $1000$ extracted bases are reviewed, the first $20$ of which are shown in Figure 14. The first basis appears to contain more random noise, while the subsequent bases shows directionality.

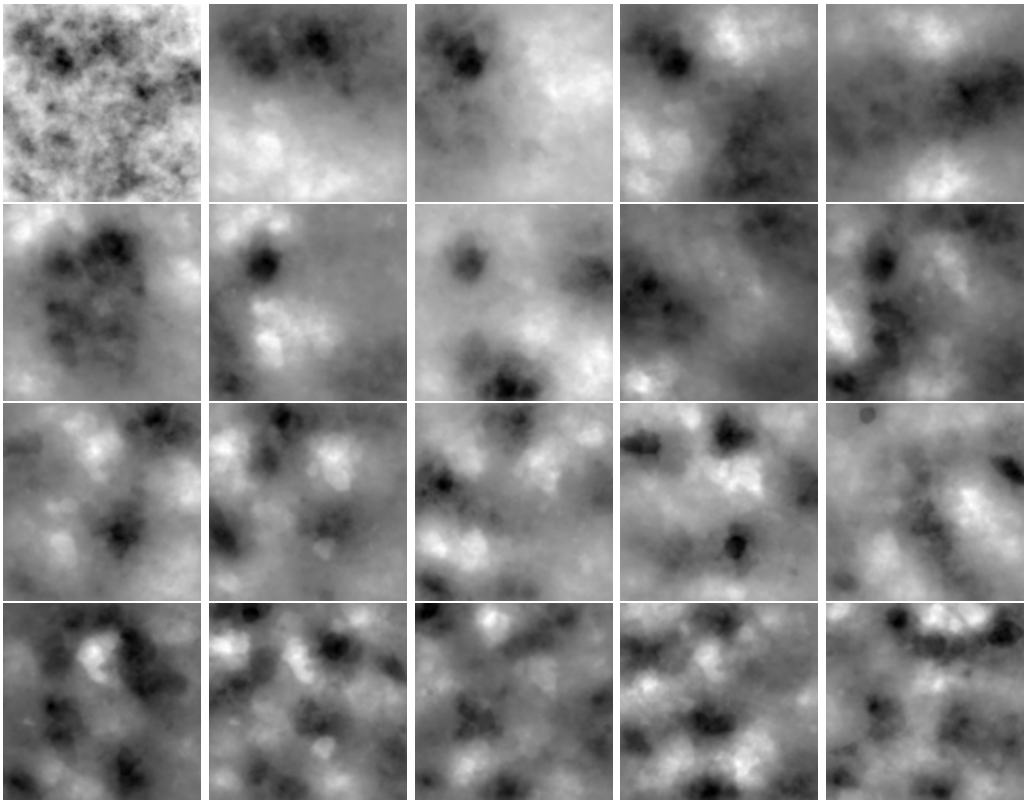


Figure 14: Results of Singular Value Decomposition on the extracted spot data. The first $20$ of the $1000$ derived basis vectors are shown.

These extracted bases are useful as a lower-dimensional representation of the ST spots for subsequent work on classifying spots, exploring spots in relation to other imaging data such as astrocytes and neurons, and eventually, comparing to the spatial transcriptomic data.

## 5.5 Variational Autoencoder

Variational autoencoders (VAEs) are unsupervised generative models that is based on an autoencoding framework [7]. The VAE can be viewed

as two coupled, but independently parameterised models: the encoder which is a recognition model and the decoder or the generative model. Through the procedure of data compression in a deep network with nonlinear activation functions, the latent vector of the VAE can serve as a non-linear low-dimensional representation of the plaque morphology data that captures the key morphological features. Here, we train a VAE on the plaque images at the location of the ST spots, which are extracted by the spot recreation technique described in Section 4.1. The same set of training and test mouse slides, described in detail in Section 9.1, is used in the training and validation of the VAE model. Each set consists of one $3$ and one $18$ months old mice, constituting a total of $2077$ and $1966$ spot images in the training and test sets, respectively.

**Preprocessing:** The extracted spots images are resized to $64 \times 64$ pixels. During training step, images in the input batch are augmented by the introduction of a random horizontal and vertical flip at a probability of $50\%$. This step is to enhance generalisation over image orientation and alleviate model over-fitting due to small training sample size.

**Model Architecture and training parameters:** A VAE model (VAE-16) with a latent dimension of $16$: an encoder with four layers of *convolution* blocks (*batch normalisation*, *LeakyReLU*) followed by two fully connected layers and a decoder with four layers of *deconvolution* blocks (*batch normalisation ReLU*), is trained with batch size $32$, at learning rate $0.001$, for $50$ epochs.

**Model performance:** The trained VAE attains reconstruction loss of $0.1553$ and $0.1876$ per pixels on the training and test set, respectively. An example of the reconstructed images in the test dataset is shown in Figure 15.

To evaluate the information captured by the VAE-16 latent space, the latent representation of spot images is visualised with their key biological (mouse age in months:Age) and image (plaque index, maximum and mean pixel values in spot image) features. To facilitate the graphical representation, the $16$-dimensional latent vector of the VAE-16 encoded spot images is projected to their first two principle components in Figure 16.

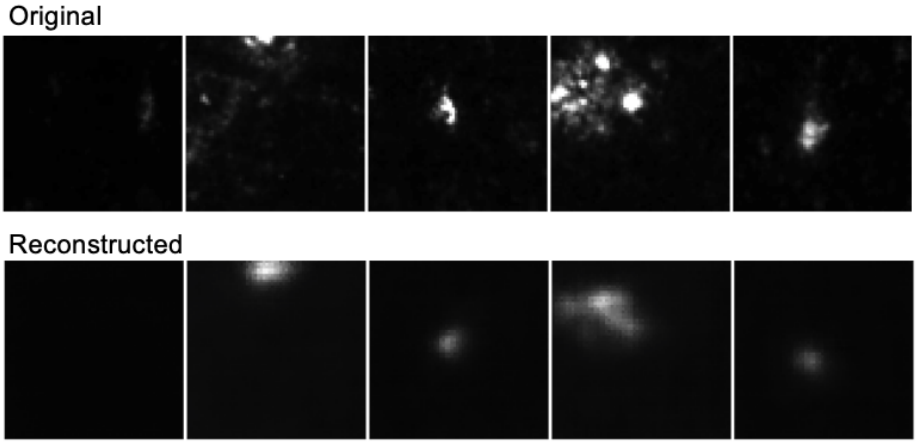



Figure 15: Reconstructed test set images by the VAE-16 model.

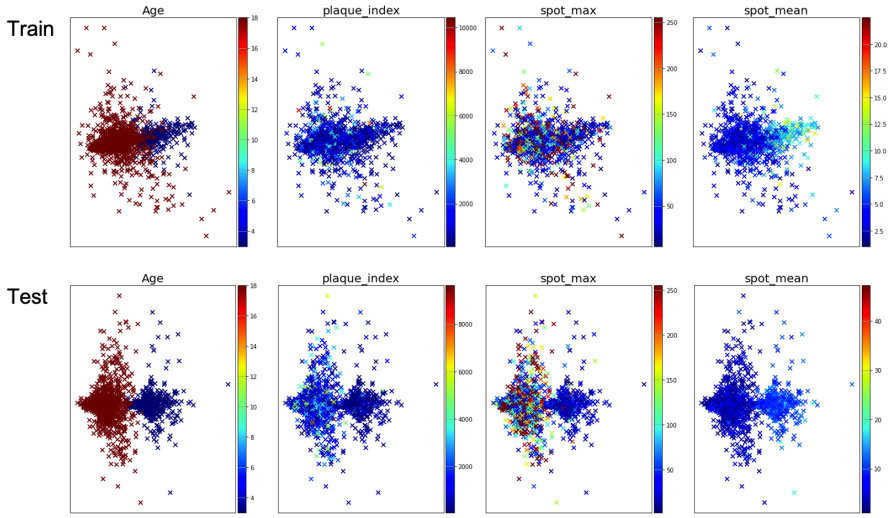



Figure 16: The VAE-16 encoded spot images represented in the first two principle components, coloured by mouse age, plaque score, maximum and mean pixel value of spot images.

The projection of the VAE-16 latent representation of spot image to it's largest variance (first principle component) is able to differentiate AD mouse age, which appears to match the variation of mean pixel values in spot images. Similar but more dispersed latent representation is

replicated in the test data, providing evidence that this observation is unlikely to be a result of an overfitted model behaviour. The VAE-16 latent vectors appear to be able to represent the key spot image features with biological relevance.

# 6 Plaque Clustering

While plaque descriptors offer intuitive metrics for analysis owing to the complex morphology of A$\beta$ plaques, some key features may be missed under this generalised procedure. The classification of plaques and plaque morphology will enable the extraction of additional discriminative features that can be directly correlated to the spatial transcriptomic (ST) information.

Previous papers have outlined three main plaque classes: core, diffuse and cerebral amyloid angiopathy (CAA) [22]. There is however no definitive categorisation for plaques. Additionally, it is difficult to ascertain as to whether only certain plaques are responsible for AD, or whether certain plaques tend to appear in certain regions of high genomic expression.

One of our primary aims in this project is to explore unsupervised plaque classification (or clustering) models based only on image data, without any additional supervised input as in [22]. After isolating the plaque images from A$\beta$ slides using Algorithm 1, the $k$-Means clustering and Visual Similarity Clustering approaches have been employed to attempt a classification of these plaques without use of any supervised knowledge.

## 6.1 $k$-Means Clustering

The $k$-means clustering optimally separates $N$ observations into $k$ convex clusters with each observation belonging to the cluster with the nearest mean centroid. This method requires the number of clusters $k$ to be pre-defined. One typically uses the elbow method to determine the best number of clusters to use, especially when it is not known how many categories data would fall in. Owing to the high dimensions of the plaque images, we have explored a more computationally efficient version of $k$-means: *Mini Batch $k$-means* [18]. Using the aforementioned elbow method with sum of squared errors of clusters to points distances, we found the optimum numbers of clusters for the plaque images to be around $4$-$5$.

## 6.2 Visual Similarity Clustering

A further improvement to Mini Batch $k$-means is attained by using a pretrained neural network for feature extraction based on visual similarity and then clustering the images using $k$-means. Once a feature vector has been extracted from the images, they can then be clustered based on how similar these feature vectors are. The pretrained model that has been used in this particular case is the VGG-16 [20] convolutional neural network (CNN), which was specifically developed for large scale image recognition. This model is used only for feature extraction, excluding the final layer responsible for prediction, and as a result, the plaque image is represented by a lower-dimensional feature vector. These feature vectors are then clustered through the use of $k$-means, allowing for the images to be clustered into $k$ natural classes. The appropriate number of clusters can be identified by plotting the within-cluster sum of squares (WCSS) distances with increasing number of clusters. The 'elbow' of the function signifies the best number of clusters. The result of this plot on our plaque image dataset is presented in Figure 17. Sharp changes in the WCSS is not observed, but the attenuation of WCSS errors starts at around *n_clusters* = $10$.

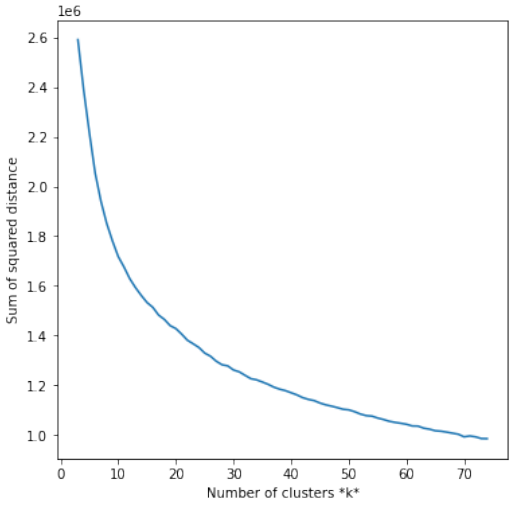
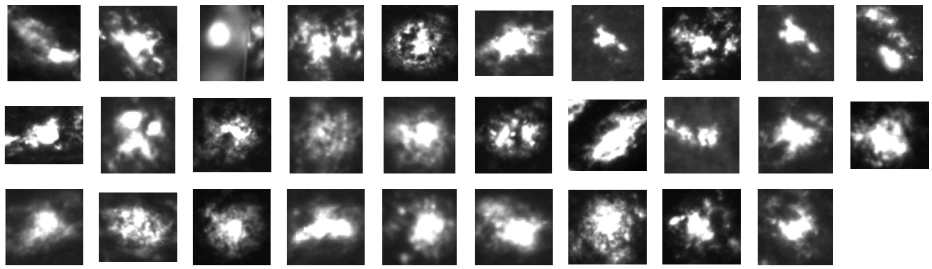
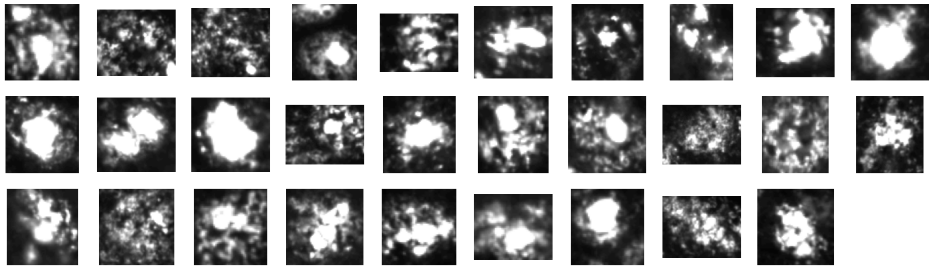



Figure 17: Elbow plot of the within clusters sum of squared distances with increasing number of clusters.

We use this minimal $10$ number of clusters to obtain 'natural' grouping of visually similar plaque images. An example of two 'visually similar' clusters is shown in Figure 18.



(a) Cluster 3



(b) Cluster 6

Figure 18: Examples of clusters extracted by Visual Similarity Clustering method.

# 7 Dimensionality Reduction of Gene Expression Data

The spatial transcriptomics data is extremely high-dimensional, featuring $N = 46,454$ normalised expression counts for each transcriptomics profile. This poses problems for a number of traditional machine learning methods, particularly as the number of features is greater than the number of data instances. Thus, it is imperative to refine the genetic data into a lower-dimensional feature set.

## 7.1 Clustering

Clustering genes with highly similar expression patterns (co-expression) into modules is performed by using a weighted gene co-expression network analysis (WGCNA) [5]. Prior to this, filtering is done on the $50\%$ most variable genes across the full library of $10,327$ ST profiles. Overall, $12$ WGCNA modules are produced, of which two ("purple" and "red") modules are the most responsive to A$\beta$ (Figure 19) [3].

## 7.2 Principal Component Analysis

Principal component analysis (PCA) can be achieved by *singular value decomposition* of the data matrix, which is preferable over the eigen decomposition of the data covariance matrix, since generation of covariance matrix may cause lost of precision. The eigen vectors are known as *principal components* and, when ordered by their corresponding eigenvalues, describe the directions in feature space that explain the most variance in the dataset. By only taking the first $M$ principal components, we can form an $M$-dimensional representation of the data, capturing most of the information in it.

While PCA is easy to compute and relatively interpretable, it has some significant drawbacks. Most of all, it is a linear method; it simply projects the data down onto a particular linear subspace of the feature space. If the data has characteristics that are not linearly separable, PCA alone will not allow us to pick these out.

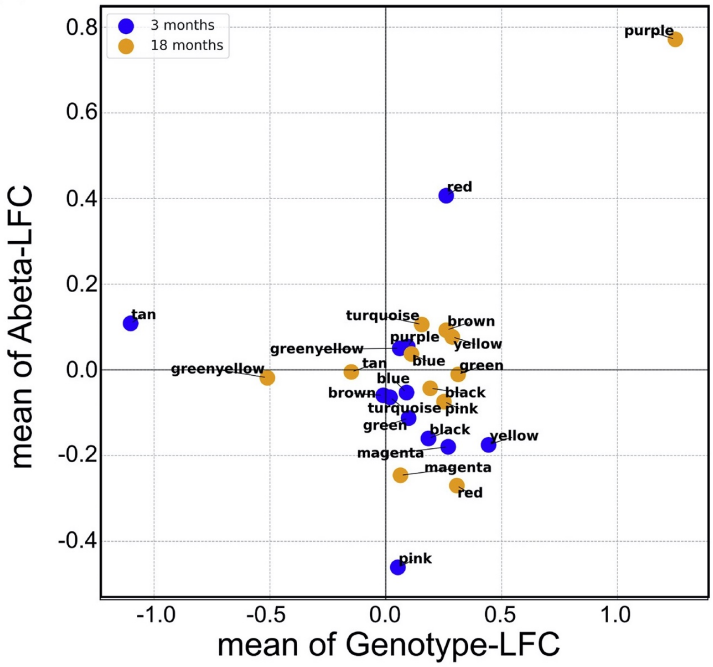



Figure 19: Co-expression networks defined by WGCNA: Summary of the differentially expressed genes in function of A$\beta$ exposure or genotype analysed by WGCNA [3].

Visual inspection of the first two components of PCA applied to the gene expression matrix shows clear capture of information regarding mouse genotype (as illustrated in Figure 20). Qualitatively, there also seems to be some structure related to the age of the mice, but there is no obvious correspondence between the labelled brain regions of the ST spots and the principal components of the gene expression data.

## 7.3 Manifold Learning

Manifold learning is based on the idea that the inherent dimensionality of data may be lower than its number of features, and that the lower-dimensional surface encapsulating the data is highly nonlinear. Machine learning methods attempt to learn the surface—or *manifold*—from the data.

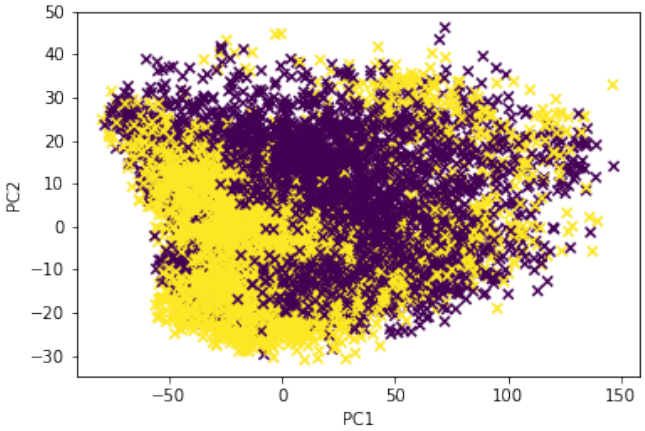



Figure 20: First two principal components of genetic data coloured by genotype.

$t$-distributed stochastic neighbour embedding ($t$-SNE) [10] is a probabilistic method for constructing the data manifold. It focuses on local structure in the data and does not require that the data come from a single, connected manifold. One notable drawback is that it is significantly more computationally expensive than the PCA.

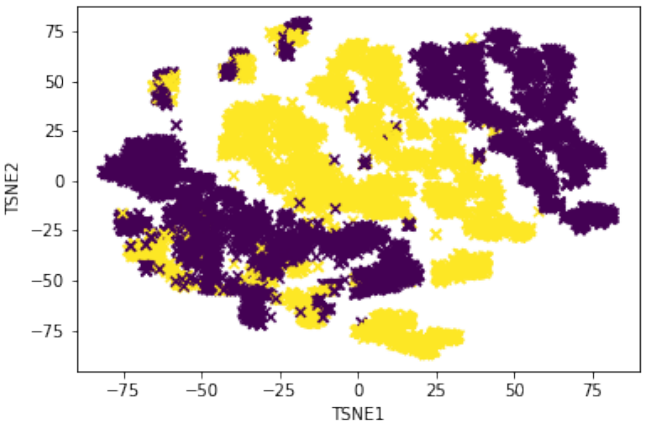



Figure 21: Two-dimensional $t$-SNE embedding after PCA preprocessing.

It was suggested during the implementation of $t$-SNE that datasets with large numbers of features (such as ours) should use PCA as a preprocessing step to suppress noise and speed up computation. We find this to vastly improve our results, both in terms of computation and time and in terms of the qualitative results of the dimensionality reduction.

$t$-SNE embedding produce much clearer visual separation of the genotypes than PCA, although it is worth noting that we have only plotted the first two principal components in the latter case (since higher dimensional visualisations are somewhat difficult).

Uniform manifold approximation and projection (UMAP) [12] is another stochastic manifold learning method. It attempts to capture global structure in the data in addition to the fine-scaled structure captured by $t$-SNE, while also giving more reproducible results and having lower computational costs.

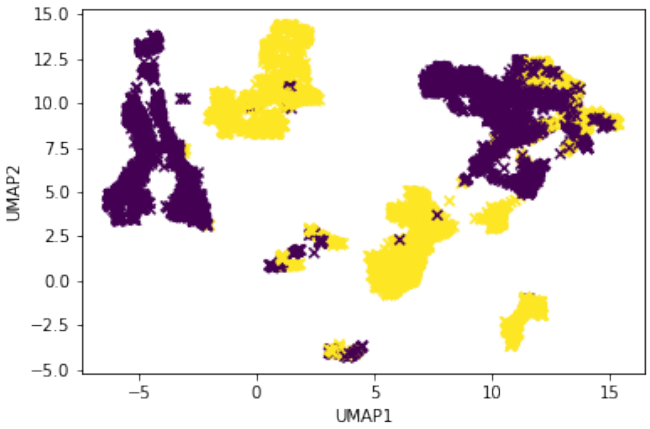



Figure 22: Two-dimensional UMAP embedding after PCA preprocessing.

UMAP's embedding also demonstrates clear separation of the genotypes. It is difficult to assess whether the global structure seen is meaningful, but the computational advantages over $t$-SNE are clear to see, with substantially reduced computational times. Once more, we find preprocessing with PCA to be advantageous.

Comparison of the three dimensionality reduction methods is illustrated in Figure 23. The PCA-UMAP appears to have attained the best separation of mouse genotype, age, and plaque index.

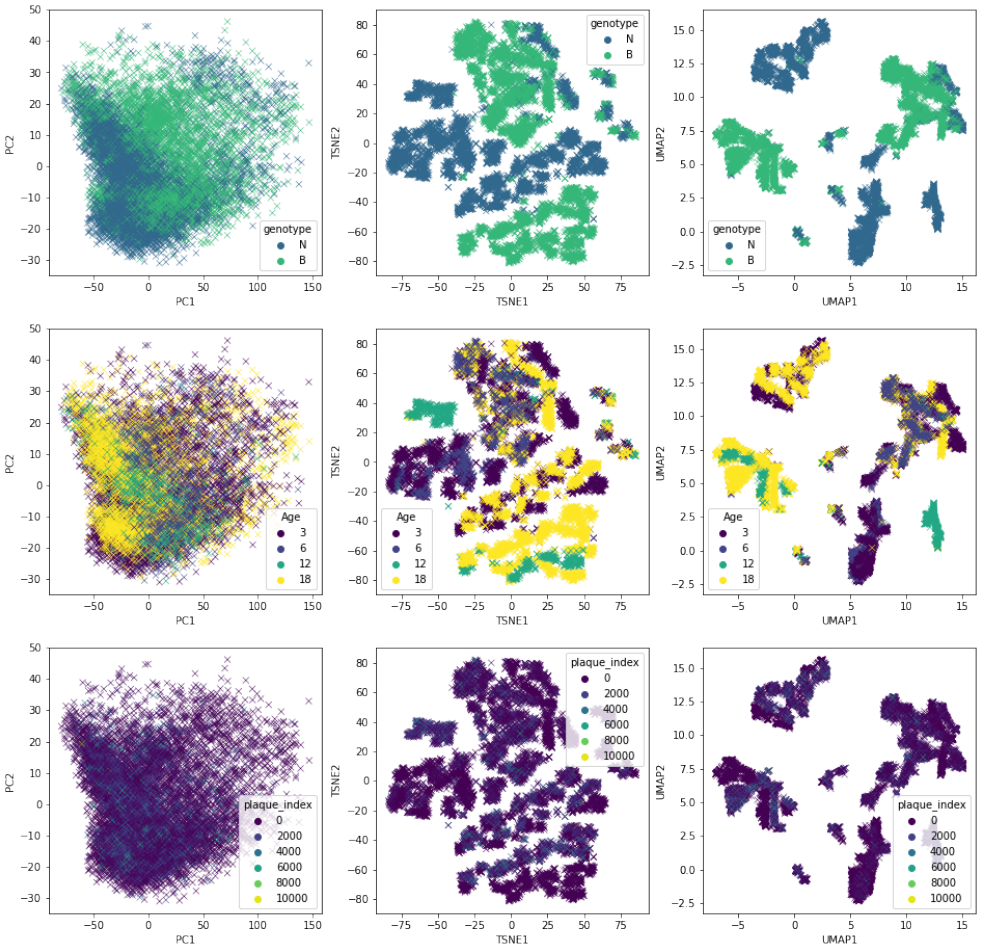



Figure 23: Comparison between the 3 dimensionality reduction methods in the organisation of the gene expression information with respect to genotype (N:AB, B:control), age in months, and plaque index.

## 7.4 Autoencoders

Autoencoders are a class of neural network architectures that attempt to *encode* their input into a lower-dimensional representation and then

*decode* it back into the original input. The better they can reconstruct their input, the better a lower-dimensional representation of the data they have learned.

The autoencoder architecture we experiment is relatively simple, with the repeating unit consisting of a linear layer followed by a ReLU activation function. We carry out ad-hoc experiments varying the depth of the network and the latent dimension (the dimension of the output of the encoder network); but we are not able to reliably reconstruct the genetic profile. Further work—including a review of the existing literature for application of autoencoders to genetic data—will be required to assess the viability of this approach.

# 8 Relationship Between Extracted Plaque and Gene Expression Features

As the first step to explore the relationship between the extracted plaque morphology and the gene expression features, a cross correlation between all extracted plaques and gene expression features is performed on the spots data. The Spearman's correlation coefficient $\rho$ across all features is displayed on Figure 24. Pairs of features with significant correlation (multiple comparison corrected using false discovery rate method) and Spearman's $\rho$ larger than $0.2$ are annotated with '+' sign. The number of features with cross-domain (plaque or gene expression) correlations are summarised in Figure 25 (red bars). The first principle component of spot images is correlated to the largest number of gene expression features, followed by the VAE-16 fourth latent vector, spot mean pixel value, and plaque index. On the other hand, the first UMAP component is correlated with largest number of plaque features, followed by gene set yellow and purple [3]. This preliminary information suggests that the spot principle component and VAE-16 latent vectors may provide improved representation of plaque morphology over the published metric - *plaque index*.

**Remark.** *This is a coarse exploration of the relationship between the plaque image and gene expression information; not all principle components are examined, and correlations between features are explored individually. Conclusion should be made separately from the prediction model described in the following section.*

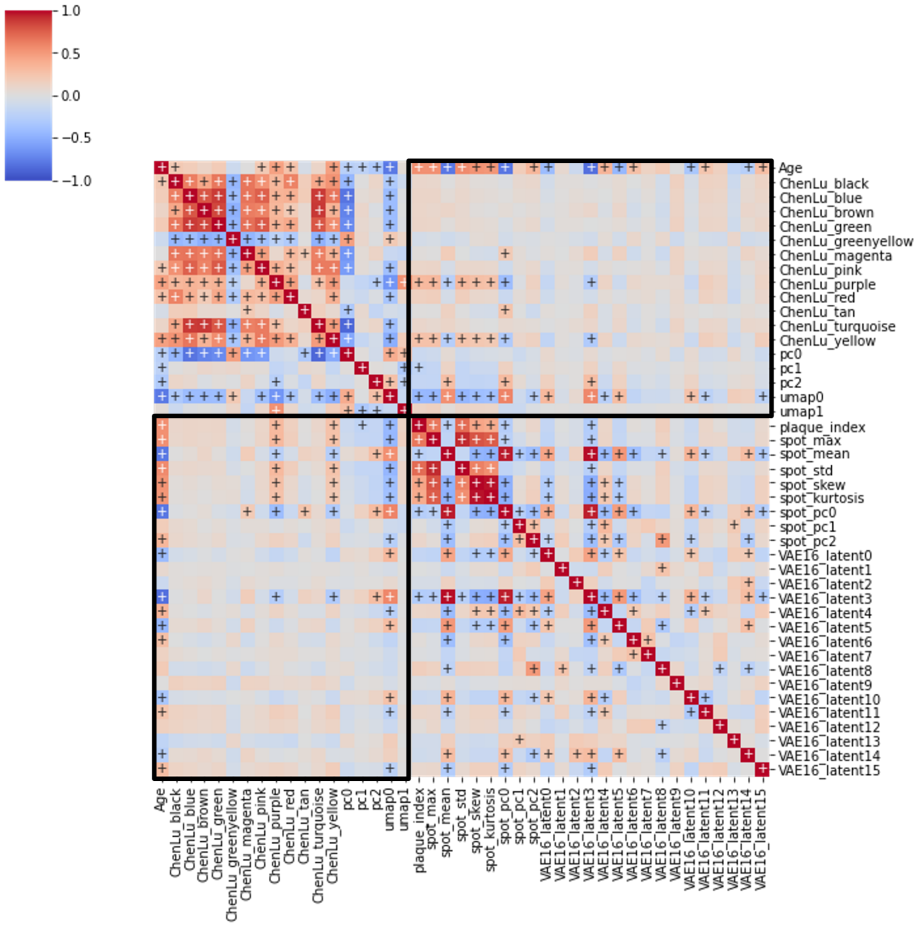



Figure 24: Cross Correlation between the extracted plaque and gene expression features.

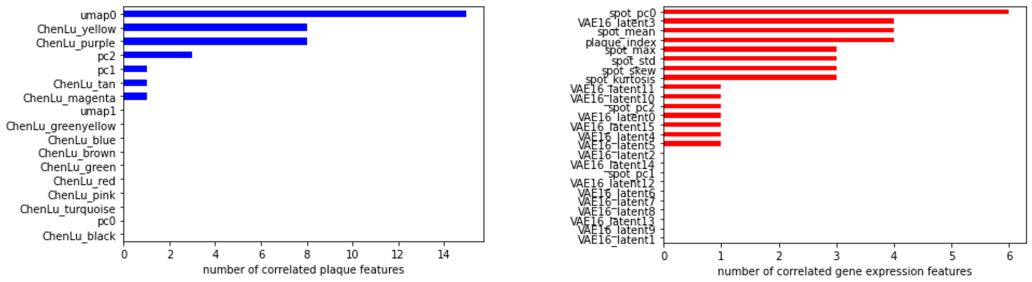



Figure 25: Ranking of gene expression and plaque features by the number of positive correlation.

# 9 Plaque Prediction Using Spatial Transcriptomics Data

## 9.1 Training/Test Split

We split the data into a training/test set for model training and evaluation. We restrict our attention to the $3$ and $18$ month old mice. It is essential not to split data from any one mouse across both the training and test sets to avoid data-leakage. In order to also have age-balanced training and test sets, we choose one $3$ month and one $18$ month mouse of each genotype, wildtype (WT) and Alzheimer's Disease (AD), to form the test set. Due to the low data availability, this results in a $50 : 50$ training/test split, which can undoubtedly affect model performance. In future studies with larger amounts of data available, it will be possible to utilise a greater percentage of the data for model training while also maintaining these properties.

Table 1: Breakdown of mice in dataset

| Mouse | Genotype | Age | Set |
|---|---|---|---|
| B03 | WT | 3 | Training |
| B04 | WT | 18 | Training |
| N02 | AD | 3 | Training |
| N05 | AD | 18 | Training |
| B02 | WT | 3 | Test |
| B05 | WT | 18 | Test |
| N03 | AD | 3 | Test |
| N04 | AD | 18 | Test |
| B06 | WT | 12 | N/A |
| B07 | WT | 6 | N/A |
| N06 | AD | 12 | N/A |
| N07 | AD | 6 | N/A |

**Remark.** *As we have made use of the provided gene modules derived from the WGCNA clustering procedure on* all *of the data, the results of the supervised learning conducted on these features cannot be directly compared to the others. They have an unfair advantage having been*

*derived from information partially contained in the test set. In order to perform a like-for-like comparison without data leakage, the clusters would have to be computed solely based on data from the training set.*

## 9.2 Dimension Reduction Methods

As demonstrated in Section 7, it is beneficial to reduce the dimension of the genetic data before applying supervised learning techniques. The dimension reduction methods we choose to carry forward are:

1. PCA with 25 components;
2. UMAP with 2 dimensions and 20 component PCA preprocessing;
3. UMAP with 7 dimensions on the raw data;
4. Mean expression level across provided WGCNA clusters[1].

Default library parameters are used in all cases.

## 9.3 Predicting Plaque Presence

Our first supervised learning problem is to classify whether or not plaques are present in the region of a spatial transcriptomics spot. At the time this analysis was carried out, the best available descriptor of this was whether the plaque score was strictly positive. As all spots from AD mice had a positive plaque score, this serves also as a proxy for the genotype of the mouse.

The methods considered are

1. Logistic regression,
2. Support vector classification (SVC),
3. $k$-nearest neighbours classification,
4. Gradient-boosted tree classification (XGBoost [2]).

Default library parameters are used in all cases.

[1] See Remark in Section 9.1 for why this is not a like-for-like comparison.

Table 2: Test set accuracy for the plaque classification task

| Dim. red. method | Log. reg. (%) | SVC (%) | KNN (%) | XGBoost (%) |
|---|---|---|---|---|
| PCA | 86.6 | ***93.8*** | **93.0** | 83.7 |
| UMAP + PCA | 46.4 | 85.6 | 86.2 | 85.6 |
| UMAP | 47.7 | 60.8 | 57.7 | 58.3 |
| WGCNA | 91.6 | **92.8** | 92.6 | 90.7 |

As shown in Table 2, the PCA and WGCNA clustering are the most effective dimension reduction methods, with both Support vector classification and $k$-nearest neighbour prediction proving to be best performed methods for predicting plaque presence. UMAP applied to the raw data performs poorly, despite its attractive visualisations in Figure 22. UMAP combined with PCA does not perform as effectively as PCA alone.

## 9.4 Predicting Plaque Score

We next attempt to predict the plaque score using a variety of regression methods. The methods considered are:

1. Linear regression,
2. Ridge regression,
3. Support vector regression,
4. Stochastic gradient descent (SGD),
5. Gaussian process regression (GPR),
6. Decision tree regression,
7. $k$-nearest neighbours regression,
8. Gradient-boosted tree regression (XGBoost).

Default library parameters are again used in all cases. Support vector, Gaussian process, and Decision tree regression are discarded after performing worse than the baseline method of simply predicting the mean plaque score from the training data for every spot. These methods may

have value in future analyses, but would first require some hyperparameter tuning to improve their performance.

Table 3: Test set Root Mean Squared Error for the regression task

| Dim. red. method | Lin. reg. | Ridge | SGD | KNN | XGBoost |
|---|---|---|---|---|---|
| PCA | 807.504 | 807.505 | 990.203 | 826.663 | 829.247 |
| UMAP + PCA | 948.817 | 948.817 | 1069.744 | 857.321 | 862.648 |
| UMAP | 978.573 | 978.572 | 1048.273 | 1050.871 | 1080.862 |
| WGCNA | **777.053** | **776.529** | 1029.625 | **777.508** | 811.516 |

Table 4: Test set Mean Absolute Error for the regression task

| Dim. red. method | Lin. reg. | Ridge | SGD | KNN | XGBoost |
|---|---|---|---|---|---|
| PCA | 418.564 | 418.564 | 409.488 | **332.950** | 363.350 |
| UMAP + PCA | 566.335 | 566.335 | 450.092 | 358.766 | 367.756 |
| UMAP | 571.644 | 571.644 | 460.751 | 490.973 | 547.354 |
| WGCNA | 426.357 | 426.050 | 427.215 | 335.965 | 389.928 |

As shown in Table 3, the WGCNA clustering is the most effective dimension reduction tool with respect to the minimisation of prediction error (evaluated as Root Mean Squared Error) and ridge regression is the most effective regression method. Tuning the regularisation strength (alpha) in the ridge regression may further improve the prediction result. PCA also demonstrates reasonable efficacy with respect to the overall prediction (evaluated as Mean Absolute Error). Similar to the classification performance, UMAP does not cluster gene expression well with respect to the plaque index evaluated at both metrics. $k$-nearest neighbour's regression in the PCA space performs the best in the minimisation of overall error. The better prediction evaluated at Mean Absolute Error may due to the nature of the algorithm, which predicts the target by local interpolation of the nearest neighbours in the training set. With reference to the mean value of plaque index in the test data at $963.27 \pm 28.01$, the magnitude of prediction error is around $30\%$ in the best working prediction methodology. Further hyperparameters tuning and data augmentation are required to achieve optimal prediction performance.

# 10 Limitations

## 10.1 Sample

The sample size of $6$ AD mice in total, with $2$ matched age group, leading to $2$ mice per training and test dataset is too small for most of the machine learning algorithms. The findings in the current report are still preliminary and warrant confirmation with larger sample size and model optimisations.

## 10.2 Plaque Feature Extraction at Spot Region

The evaluation of relationship between plaque morphological features and gene expression in this report are limited to the spot region, where plaque features are extracted from the region of spatial transcriptomic measurement location. Although this approach retains the precision in spatial transcriptomics information, feature extraction with regard to plaque morphologies are incomplete.

## 10.3 Translation of Gene Expression Information to Unmeasured Region at Plaque Location

The plaque image and transcriptomics data are extracted from different experimental systems with different resolution. Owing to the low spatial resolution of the ST data, the gene expression information cannot be directly estimated at the location of individual plaques. Due to time limitation, we have not explored methods to translate the two disparate data types into a common spatial registry. The completion of this step would provide complementary information that is lost at the 'spot centred' analysis.

# 11 Main Conclusions

The main conclusions of our current study are five-fold:

**1. Plaque image extraction:** We have established a working procedure for plaque ROI extraction from the A$\beta$ immunofluorescent brain image using conventional computer vision methods: blurring, thresholding, and segmentation.

**2. Plaque feature extraction:** We have produced promising lower-dimensional representation of plaque morphologies that appears to outperform the previously established plaque index in encapsulating biologically relevant information. This representation is the first principle component and the fourth latent vector of VAE that has been trained on the spot images.

**3. Gene expression feature extraction:** A detailed analysis between various dimensionality reduction methods on the gene expression data is executed, showing that PCA performs favourably compared to the $t$-SNE, UMAP, AE, and the original WGCNA method.

**4. Relationship between plaque image and gene expression features:** Under default model hyperparameters, support vector classification on the PCA represented gene expression best predicts plaque presence. Ridge regression on WGCNA represented gene expression performs best on predicting plaque index in terms of the minimisation of prediction error. $k$-nearest neighbours on PCA embedded gene expression results best prediction on plaque index in terms of overall error.

**5. Key significance:** We have explored a number of methodologies to extract key features from plaque morphologies and transcriptomics data. This allows future efforts to employ more explainable machine learning models, e.g. regression, tree-based models, VAE, etc., to learn the relationship between the extracted plaque morphologies and the transcriptomics information.

# 12 Future work

## 12.1 Analysis of Plaque Image Data

The visual similarity clusters of plaque groups illustrated in Figure 6.2 can be used to label and count the number of plaque groups in the spot region, which in turn can serve as input features in a machine learning (ML) model. In addition, a translated gene expression information at the plaque location can be used as the input features for the prediction of plaque clusters, coefficients of which may reveal the relationship between visually identified plaque groups and transcriptomics variation.

## 12.2 Analysis of Spatial Transcriptomics Data

Another candidate dimensionality reduction method that is not investigated during the course of this project is Latent Dirichlet Allocation (LDA). This modelling approach attempts to explain observation by the presence of unobserved groups. It has already been widely applied in population genetics and could be an interesting avenue of research for this type of data.

## 12.3 Plaque Prediction

The most straightforward next step of the analysis is to use the correlated plaque or gene expression features revealed in Section 8 to build an explainable machine learning model, e.g. regression or tree-based models. The regression coefficients or features importance in the models can then be used to evaluate the relationship between gene expression and plaque morphology.

In terms of ML methodologies for plaque prediction, there can be a number of improvements. Firstly, kernelised versions of ridge regression and PCA could offer greater flexibility than the standard variants examined in this report. Secondly, there are methods that naturally combine the dimension reduction and classification/regression problems into a single step. Examples of these are partial least squares regression (which operates much like PCA while also ensuring that the chosen directions correlate well with the response) and supervised UMAP.

# 13 Team members

## Participants

**Aaron Wagen** is a neurologist undertaking a PhD at the Institute of Neurology at University College London. He is exploring spatial transcriptomics of Parkinson's disease, and interested in computational approaches that allow exploration across systems and scales.

**Bertrand Nortier** is a postdoctoral research fellow at the University of Exeter. His interests include computational statistics and statistical machine learning.

**David Jarrett** is undertaking his MSc in Data Science at Newcastle University. His interests include machine and deep learning.

**Hamza Alawiye** is a Heilbronn Research Fellow at the University of Bristol. His interests include mathematical modelling, Bayesian machine learning and scientific computing.

**Kevin Wang** is a researcher in the field of applied and theoretical machine learning. He is currently finishing his undergraduate degree in computer science at the University of California, Berkeley.

**Kulsoom Abdullah** is a data scientist with 7 years of experience in data science and machine learning; currently working in healthcare and actively transitioning to applied ML and deep learning research. She attended the University of Central Florida and received her doctorate in Electrical/Computer engineering at the Georgia Institute of Technology. In spare time, she is a competitive Olympic Weightlifter.

**Phazha Letlhogonolo Kevaun Bothongois** a PhD Candidate in Neuroscience at Queen Mary University of London. Her interests are in ethnic and socioeconomic determinants of dementia.

**Sebastian Mararu** is a computer scientist with a master's in Bioinformatics and Systems Biology from the University of Manchester. He enjoys sifting through dirty data and trying to make sense of it, especially in a biological context.

**Vidya Ganesh** has recently finished undergraduate in electronics and communication from Vellore Institute of Technology, India. She is currently a project associate at Indian Institute of Technology, Madras.

## Principal Investigators

**Abhirup Banerjee** is a Senior Postdoctoral Researcher in the Division of Cardiovascular Medicine, Radcliffe Department of Medicine, University of Oxford, and also in the Institute of Biomedical Engineering, Department of Engineering Science, University of Oxford. His research interests include Biomedical Image Analysis, Machine Learning, Artificial Intelligence, Statistical Pattern Recognition, Image Processing, and so forth.

**Yeung-Yeung Leung** is a Postdoctoral Researcher at Imperial College London, Brain Science Institute. Her PhD is in Computational Neuroscience and her expertise lies within explainable machine learning methods, bioinformatics, time series and image data analysis.

## Challenge Owners

**Mark Fiers** is a Bioinformatics staff scientist at the Flemish Institute of Biotechnology in Leuven, Belgium and at UCL London focusing on understanding the role of amyloid beta during the onset of Alzheimer's Disease.

**Diego Sainz** is a Bioinformatician at the Flemish Institute of Biotechnology in Leuven. His interests are applying computational methods to solve real life problems as Alzheimer's Disease and understanding its context in the human brain.

# Acknowledgement

We would like to thank the Alan Turing Institute as well as the Challenge Owners for their support and the computational resources provided. We are also grateful to Sebastian Mararu and Aaron Wagen (Project Facilitators), Abhirup Banerjee and Yeung-Yeung Leung (PIs) for their help and advice on the technical aspects of the project and project management.


# References


[1] Gary Bradski and Adrian Kaehler. “OpenCV”. In: *Dr. Dobb’s journal of software tools* 3 (2000).

[2] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. CA, USA: ACM, 2016, pp. 785–794.

[3] Wei-Ting Chen et al. “Spatial transcriptomics and in situ sequencing to study Alzheimer’s disease”. In: *Cell* 182.4 (2020), pp. 976–991.

[4] fmw42. *answer to: Drawing bounding rectangles around multiple objects in binary image in python*. https://stackoverflow.com/questions/63923800/drawing-bounding-rectangles-around-multiple-objects-in-binary-image-in-python. 2020.

[5] Steve Horvath. *An overview of weighted gene co-expression network analysis*. 2005.

[6] Jiwon Jeong. *Computer Vision for Beginners: Part 4 Contour Detection and Having A Little Bit of Fun*. https://towardsdatascience.com/computer-vision-for-beginners-part-4-64a8d9856208. 2019.

[7] Diederik Kingma and Max Welling. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends in Machine Learning* 12 (2019), pp. 307–392.

[8] Eliyaz KL. *answer to: how to add border around an image in opencv python*. https://stackoverflow.com/questions/36255654/how-to-add-border-around-an-image-in-opencv-python. 2020.

[9] Ed S. Lein et al. “Genome-wide atlas of gene expression in the adult mouse brain”. In: *Nature* 445 (2006), pp. 168–176.

[10] Laurens van der Maaten and Geoffrey E. Hinton. “Visualizing Data using $t$-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[11] Leland McInnes, John Healy, and Steve Astels. “hdbscan: Hierarchical density based clustering”. In: *Journal of Open Source Software* 2.11 (2017), p. 205.

[12] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. arXiv: 1802.03426 [stat.ML].

[13] Neil Muller, Lourenço Magaia, and Ben M Herbst. “Singular value decomposition, eigenfaces, and 3D reconstructions”. In: *SIAM review* 46.3 (2004), pp. 518–545.

[14] NumPy. *Singular Value Decomposition*. https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html. 2021.

[15] Nobuyuki Otsu. “A Threshold Selection Method from Gray-Level Histograms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.

[16] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[17] Rowayda A Sadek. “SVD based image processing applications: state of the art, contributions and research challenges”. In: *arXiv preprint arXiv:1211.7102* (2012).

[18] SciKit. *Mini Batch $K$-means*. https://scikit-learn.org/stable/auto_examples/cluster/plot_mini_batch_kmeans.html. 2021.

[19] Dennis J Selkoe and John Hardy. “The amyloid hypothesis of Alzheimer’s disease at 25 years”. In: *EMBO molecular medicine* 8.6 (2016), pp. 595–608.

[20] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[21] Balázs Szalontai et al. “SVD-clustering, a general image-analyzing method explained and demonstrated on model and Raman micro-spectroscopic maps”. In: *Scientific reports* 10.1 (2020), pp. 1–15.

[22] Ziqi Tang et al. “Interpretable classification of Alzheimer’s disease pathologies with a convolutional neural network pipeline”. In: *Nature communications* 10.1 (2019), pp. 1–14.

[23] open cv team. *Cropping an image using opencv*. `https://learnopencv.com/cropping-an-image-using-opencv/`. 2020.

[24] open cv team. *tutorial py contour features*. `https://docs.opencv.org/master/dd/d49/tutorial_py_contour_features.html`. 2020.

[25] Matthew Turk and Alex Pentland. “Eigenfaces for recognition”. In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.

[26] OpenCV-Python Tutorials. *Different methods of thresholding*. `https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html`. 2020.

turing.ac.uk
@turinginst