Single-nucleus profiling of human dilated and hypertrophic cardiomyopathy

By performing single-nucleus RNA sequencing on nearly 600,000 nuclei in left ventricle samples from 16 non-failing hearts and 11 hearts with dilated cardiomyopathy and 15 hearts with hypertrophic cardiomyopathy, this study identifies extensive molecular alterations in failing hearts at single-cell resolution. At the tissue and cell-type level, the transcriptional patterns of hearts with dilated or hypertrophic cardiomyopathy converged broadly. To assess this transition of fibrotic cells, the scientists used a CRISPR-knockdown screen in primary human cardiac fibroblasts; knockout of certain fibroblast transition-related genes reduced the myofibroblast cell-state transition in response to TGF-1 stimulation. The findings offer novel potential therapeutic targets and heart failure biomarkers as well as insights into the transcriptional diversity of the human heart in health and illness.

Left ventricle (LV) samples from 44 people, including 12 with dilated cardiomyopathy (DCM), 16 with hypertrophic cardiomyopathy (HCM), and 16 with non-failing (NF) hearts, were subjected to single-nucleus RNA sequencing (snRNA-seq) in duplicate. They discarded 8 samples after applying severe quality control, leaving at least 1 technical replicate from 11 DCM, 15 HCM, and 16 NF LVs. All DCM and HCM patients exhibited advanced cardiomyopathy necessitating transplantation. For patients with DCM, the LV ejection fraction (LVEF) was less than 20%, but for individuals with HCM, the LVEF was less than 50% for 7 patients and 50% or more for 8 patients.

The remaining 592,689 nuclei then grouped into 21 clusters based on transcriptional similarities after the low-quality nuclei were excluded. On the basis of gene ontology10,11, genes that were expressed selectively in each cluster were identified, and their enrichment in biological processes was evaluated.

The first graph I plotted in my project corresponds to the dendrogram demonstrating the similarity of the above-mentioned cluster centroids which were aggregated based on transcriptional similarity.

This first dendrogram plot shows hierarchically clustered samples which had transcriptional similarity and a dendrogram is plotted in order to show their similarity.

Data are grouped into groups in a tree structure in a hierarchical clustering method. Every data point is first treated as a separate cluster in a hierarchical clustering process. The following steps are then repeatedly carried out by:

Choosing the two clusters that are the most similar to one another, then merge them. These procedures must be repeated until all the clusters are combined.

The goal of hierarchical clustering is to create a hierarchy of nested clusters. a Dendrogram, a type of graph depicts this hierarchy graphically and is an inverted tree that explains the sequence in which elements are combined (bottom-up view) or clusters are dispersed (top-down view). A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits

In order to achieve this I used libraries like pandas, scipy, seaborn, and matplot. I first read the data which was in the form of a csv file using the function pd.read_csv(_) which uses the pandas library. Then I created a list of all the columns in my dataframe to go ahead with the further steps in order to achieve the centroid values of the clusters required for plotting. The columns that were present in my data frame were 'Names', 'X', 'Y', 'Category' out of which I wanted just X, Y and category which is I popped out the names column and created a new variable to store my dataset without the names.

Next I create I choose all the unique categories and store it in a new variable to retain the original data frame. Every value in the data frame is converted to numeric values using the command .apply(pd.to_numeric) so that I have all the numerically quantified data to plot the dendrogram. The next step is concatenating the unique category data frame and the data frame where values were converted to numeric values to get the final data frame required for plotting with Xmean and Ymean values along with unique cell type categories.

Then iterating is performed using a for loop through the clusters to calculate mean of X and Y as we need to obtain the centroid of the clusters and plot the heatmap as per the paper. After this index values are moved to the data frame and the final data frame has index values, x, y values and the unique cell types.

In order to go ahead with the plotting I created a numpy array of all the x mean and y mean values and spliced the array in order to remove the categories and retain just the mean values (numerical values). For this I used the command: data.to_numpy(copy = True) and

array[:,0:2] to splice the array. Dendrogram was then plotted for the numpy array containing the x mean and y mean values.

The second plot is a heatmap representing Predicted expression of genes showing interesting patterns based on a negative binomial generalized additive model for each gene across pseudo-time in patient and patient. From the heatmap produced it was observed that Several genes, including *SLC44A5*, *COL22A1*, *POSTN*, *AEBP1*, *JAZF1* and *THBS4* showed increased expression across the trajectory, whereas other genes such as *NEGR1*, *PDGFRA*, *C7*, *FBLN5* and *COL4A4* showed decreased expression across the trajectory

In order to obtain this plot I used libraries such as csv, pandas, seaborn, numpy and matplot.

I first open the csv file using the command with open csvfile and read the csv file into csv_reader where I create a list of my data frame. I also declare a variable first = True that tell us when we read the 1st row and when we don't as it contains just the names of the cell types.

Then iteration was performed through every row in the data frame using a for loop in order to get the data required for plotting. Here I used decision making using if else statement condition being first = True which means that first row containing columns is being read. If that statement is true it saves first part of 2nd row to sample_names which is used for plotting . Else the genes list is appended with the first item of the first row that contains the gene names and data list is appended with the rest of the rows in the data which contains the binomial generalized additive values corresponding to the gene expression levels. Then a numpy array for the dataframe and all the values are converted to float to achieve the plot.