Ln 2: Imports everything necessary

Ln 3: Changes some display settings

**Pre-Processing Data**

Ln 4: imports data file and converts it into a pandas dataframe.

Ln 7: Takes values out of dataframe, multiplies them by 1000, takes a log and adds 1. expr\_mat is currently a dataframe, expr\_mat\_log\_t is dataframe after log scaling.

Ln 8: expr\_reduced is a dataframe containing the principal component analysis to reduce the dimensionality to 20 of the transpose of the log scaled data.

Ln 9: joint\_TSNE is a randomly initialised TSNE fit to the expr\_reduced data.

Ln 10: adata is an annotated matrix of the principal component analysis of the dataframe. A neighbourhood graph is calculated on it, and it’s communities are computed using Louvain’s algorithm with a resolution of 0.2. It puts all the Louvain communities into a list, and converts them to integers.

Ln 11: Finds all the unique elements of the list of Louvain communities.

Ln 12: Plot a graph of the joint tsne, size of 3, one colour for each of the communities and blending value of 0.8

Ln 13: Method definition which takes an expression matrix, a prop and a random seed as arguments. It seeds the random bitgenerator with 0, and sets n\_rows and n\_cols to the corresponding values of the expression matrix. n\_elem is the rounded value for the prop\* col (expression matrix). Uses arrange from numpy to generate an array with the indices of n\_cols.

Then for each row, it randomly shuffles the columns amongst themselves.

Ln 14: Converts Louvain cluster array back into an array

Ln 15: selected\_cells is an array of the clusters we want. expr\_matsub is a sub-matrix of of the expression matrix with only the data for the selected cells. Selected columns of expr\_matsub are then shuffled using the method defined in 13.

A copy of the expression matrix is then made. Another called expr\_ matrix stores the selected rows in itself, which is now noisy data.

Ln 16: shows difference in shape of noisy data and selected cells.

Ln 17: Repeat line 7 with noisy data instead of original

Ln 18: Colour loading

Ln 19: Plot noisy data with same parameters as in line 12.

**ATAC-seq**

Ln 20: Loads the data tsv as a dataframe using pandas

Ln 21: Performs a principal component analysis on the dataframe for to get the 10 most important components, then fits and transforms the data of the transpose of the dataframe to the PCA.

Ln 22: shows shape of data after PCA.

Ln 23: fits and transforms a randomly initialised TSNE to the PCA’d data and plots it with the above parameters

**Random Modality**

Ln 25: Generates a random 1047\*20 matrix

Ln 26-27: fits and transforms a randomly initialised TSNE to the noise matrix and plots it with the above parameters

**Joint Visualisation**

Ln 28: data is a dictionary storing expr\_data at “rna” and atac\_reduced at “chromatin”.

Ln 29: joint\_tsne\_obj is a randomly initialised joint TSNE, which is then fit and transformed to the data dictionary and stored in joint\_tsne.

Ln 30: imports joint\_metrics library, then computes KNI and CARI metrics.

Ln 31: Plots the joint modality model for the reduced noise matrix of rna and the reduced atac matrix for chromatin according to the constraints given above.

**Joint 3 Modalities**

Repeats the same procedure as above with the third modality being the random noise matrix computed above.