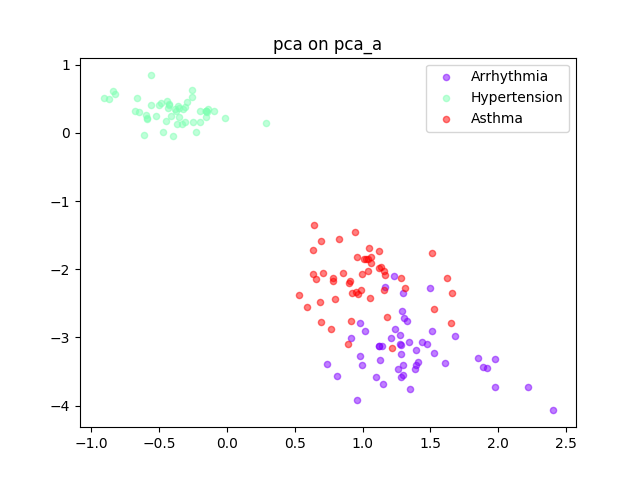
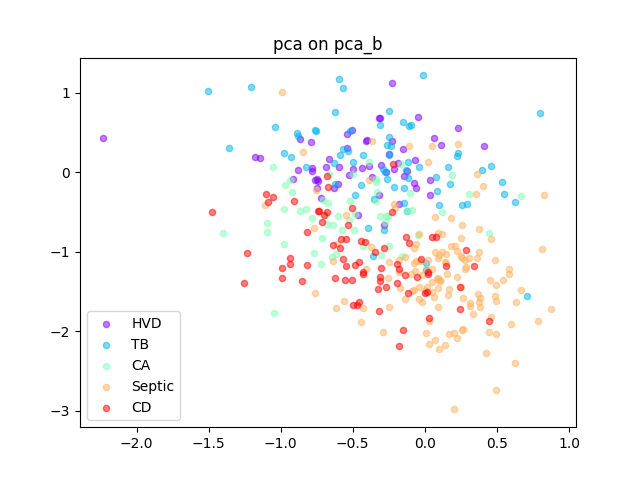
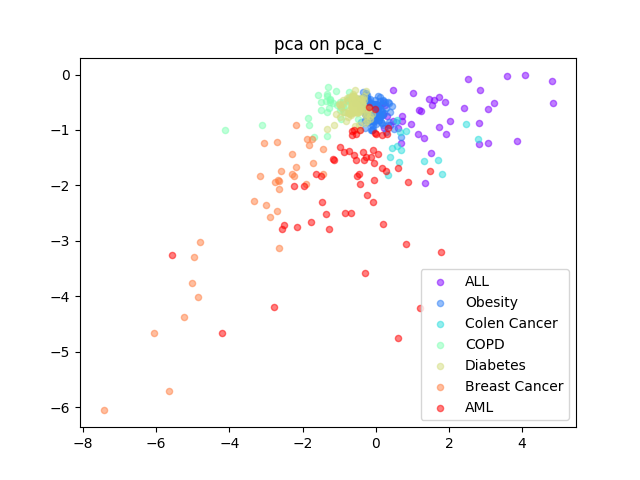
PCA ALGORITHM

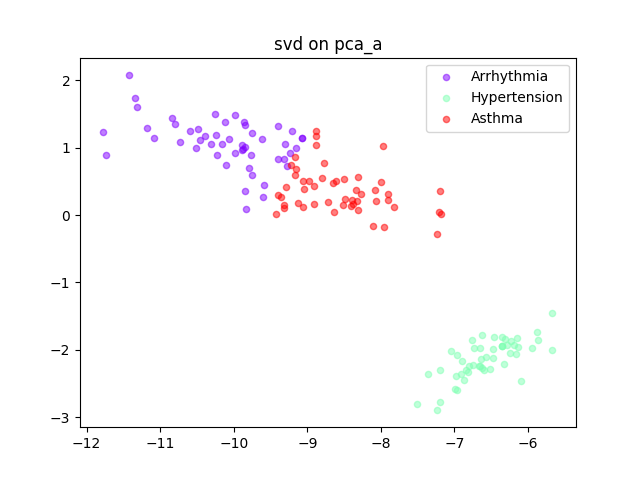
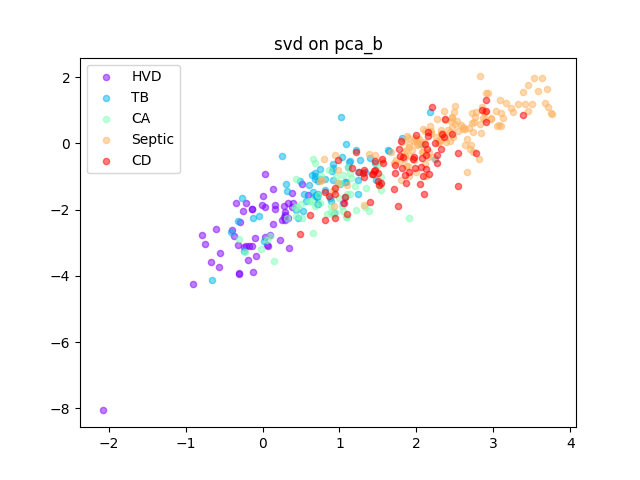
# PLOTS on datasets

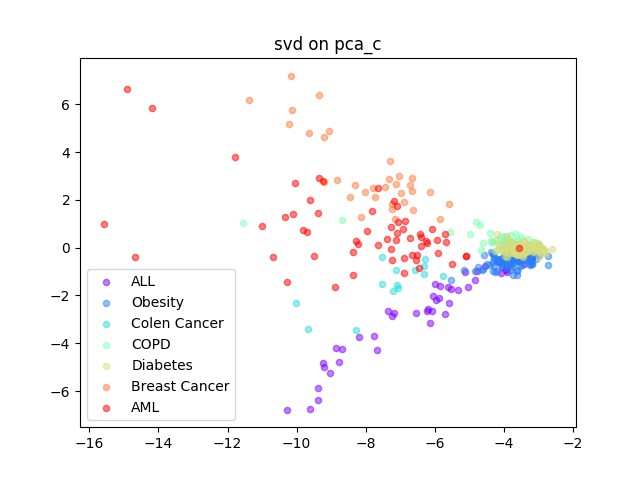
1. scatter plots on PCA

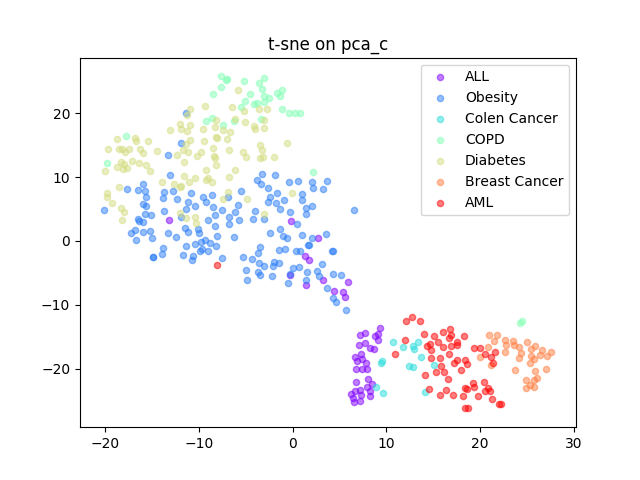
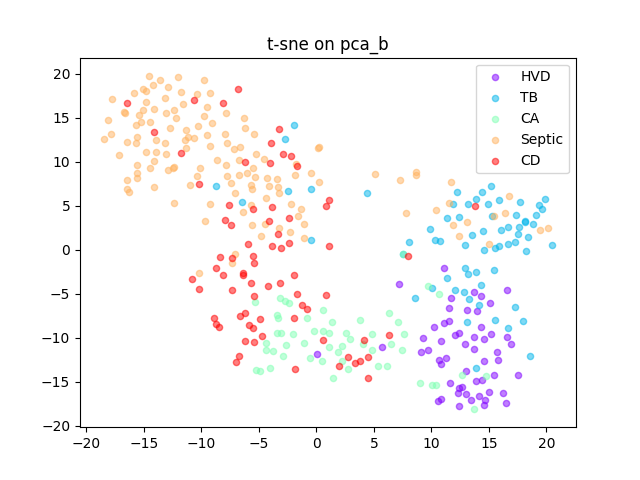
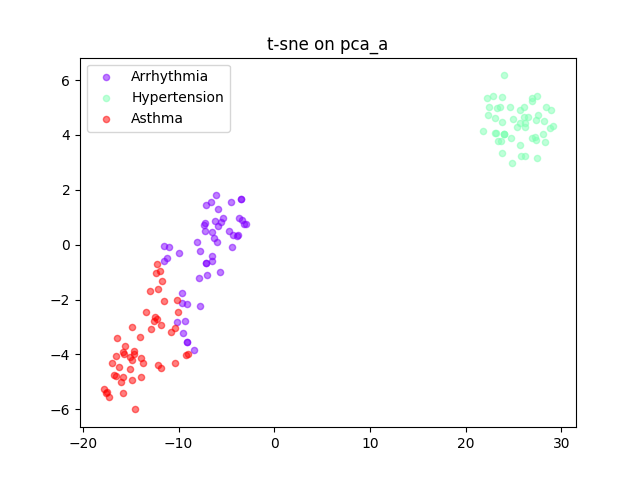


1. scatter plots on SVD



1. scatter plots on T-SNE



# PCA implementation detail

I implement the PCA algorithm by these steps.

First, load dataset and separate it to get a feature matrix and a label vector.

Second, calculate the mean vector on the matrix, then adjust the original matrix by the mean vector. Here I use the broadcasting idiom of Numpy to simplify the code.

Third, compute the covariance matrix S of the adjusted matrix, then do eigenvectors and eigenvalues decomposition on the covariance matrix.

Last, using the indexs of first two largest eigenvalues to get the corresponding eigenvectors, pack them to matrix and use it to do the dot product on original matrix.

Followed these steps, finally we can get a two dimensions matrix.

# Results discussion

From the scatter plots, I find all three algorithms PCA, SVD, T-SNE separate the pca\_a data pretty well. But for the pca\_b dataset, we can see the boundary among classes is not so clear among all three classes. And for the pca\_c dataset, the pattern of PCA and SVD on dimension reduction seems a bit similar, both of them gather the diabetes and obesity class tightly. But for T-SNE, although the distance of points between diabetes and obesity class is close, but compared to PCA and SVD, they can be separated more easily.