Data Mining and Bioinformatics

PCA Report

Saurabh Bajoria 50208005
Vidhi Shah 50207090
Sucheta Mulange 50206855

Dimensionality Reduction:

Dimensionality Reduction is used to reduce the number of features by choosing the most important ones that still represent the entire dataset. One of the main advantages of Dimensionality Reduction is that the algorithms improve the results of classification task.

There are two principal algorithms for dimensionality reduction: Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA).

PCA:

The idea behind PCA is to find a low-dimension set of axes that summarizes data. PCA performs a linear transformation moving the original set of features to a new space composed by principal component.

The eigenvectors represents the new set of axes of the principal component space and the eigenvalues carry the information of quantity of variance that each eigenvector have. So, in order to reduce the dimension of the dataset we are going to choose those eigenvectors that have more variance and discard those with less variance. The eigenvector with the highest eigenvalue is therefore the principal component.

PCA is a useful way to 'compress' data: the output components behave just like normal dimensions, but they are an orthogonal linear combination of the original features, preserving decreasing amounts of the original variance.

The conventional method for calculating PCA requires us to compute the full covariance matrix, and so it suffers from memory complexity and can be numerically unstable.

Steps to perform the Algorithm:

1. Calculate the mean vector (taking the mean of all rows)
2. Adjust the original data by the mean:
   X' = X – M
3. Compute the covariance matrix S of adjusted X
   S=XX.T/(n-1)
   Where n= number of rows
4. Find the eigenvectors and eigenvalues of S using numpy's eig function
5. Arrange the eigen vectors with decreasing value of eigen values using numpy's argsort
6. Choose the top 2 eigen vectors with highest eigen values
7. Generate a new set of values yi by linear combination of eigen vectors and adjusted dataset
8. Plot the above generated 2-d set of points to get a clustered graph.

SVD:

It is possible to perform PCA by using Singular Value Decomposition which is a purely numerical technique based on simple linear algebra - thus more stable and can be optimized for speed.

An arbitrary matrix A with m rows and n columns is converted into a feature reduced version with m rows and k columns where k,m.

Amn=UmmSmnV.TnnAmn=UmmSmnVnnT

Amk=UmmSmkVTkkAmk=UmmSmkVkkT

Amk retains as much variance as possible from the original dataset A, whilst using fewer features. Thus we can choose to store less data and use simpler models.

The operation(steps to implement PCA) is expensive when X is very large or when X is Very small. So the best way to compute principal components is by using SVD.

TSNE:

t-Distributed Stochastic Neighbor Embedding (t-SNE) is another technique for dimensionality reduction and is particularly well suited for the visualization of high-dimensional datasets. Contrary to PCA it is not a mathematical technique but a probabilistic one.

This means that it looks at the original data that is entered into the algorithm and looks at how to best represent this data using less dimensions by matching both distributions. The way it does this is computationally quite heavy and therefore there are some (serious) limitations to the use of this technique.

t-SNE relies on severe parameters : perplexity, early exaggeration, learning rate, number of iterations - though default values usually provide good results. rough pca worked well while rough t-sne gave me random looking results.

PCA, SVD and t-SNE:

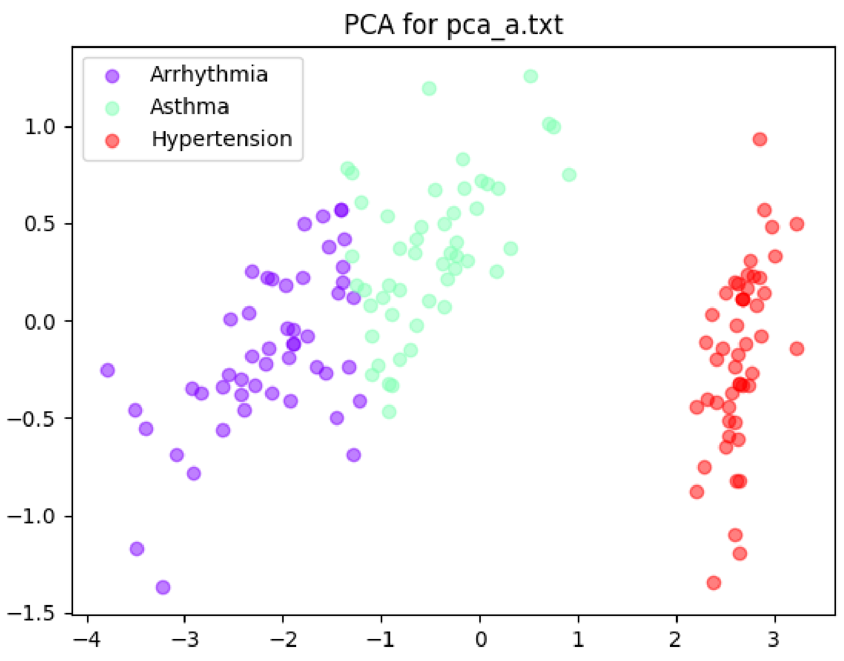
PCA performs a linear transformation moving the original set of features to a new space composed by principal component. It computes Covariance matrix, Eigen Vectors and Eigen Values for the transformation.

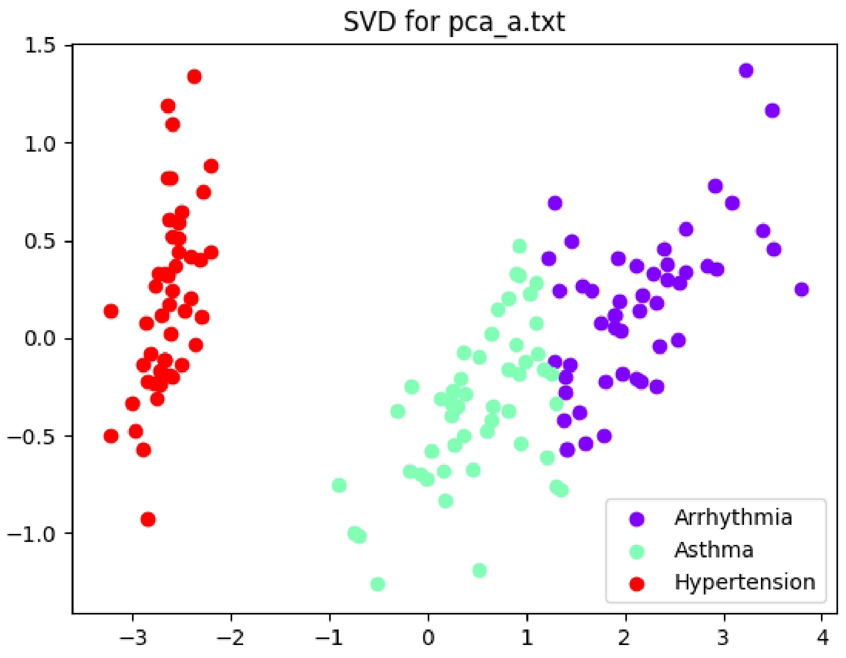
SVD is purely a mathematical technique. It factors an m x n matrix, M, of real or complex values into three component matrices, where the factorization has the form USV*.

PCA can only capture linear structures in the features. Whereas, the t-SNE algorithm works in a very different way and focuses to preserve the local distances of the high-dimensional data in some mapping to low-dimensional data.

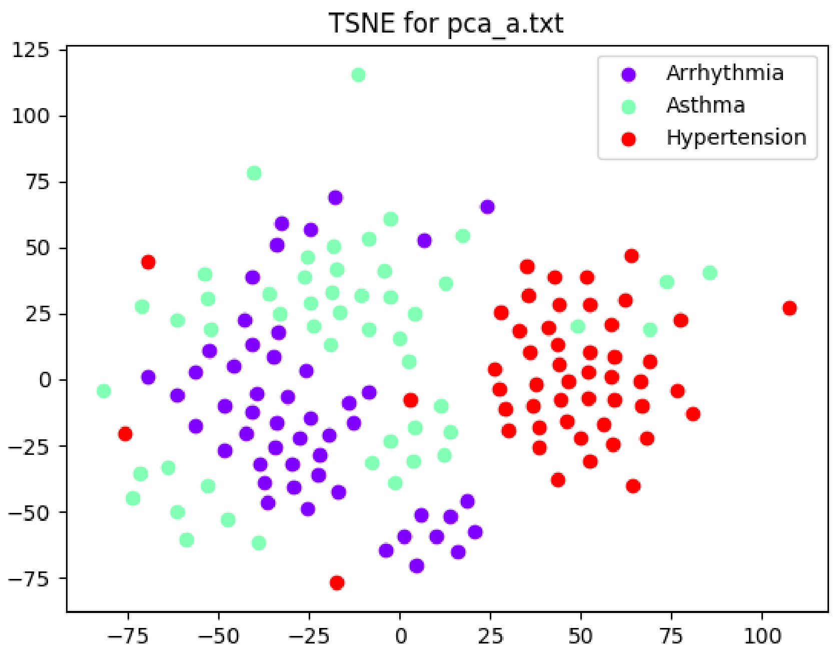
Output:

1. pca_a.txt



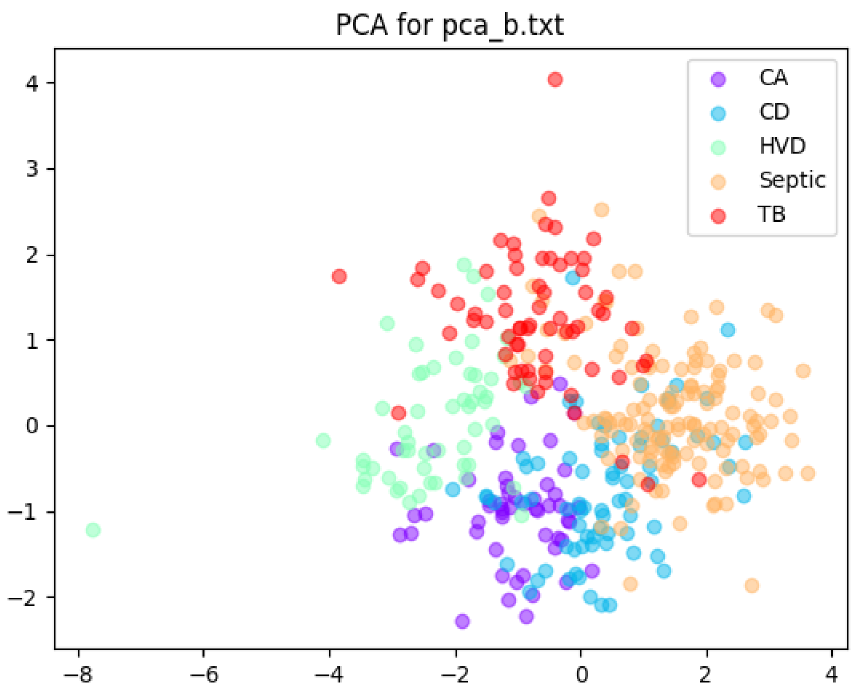


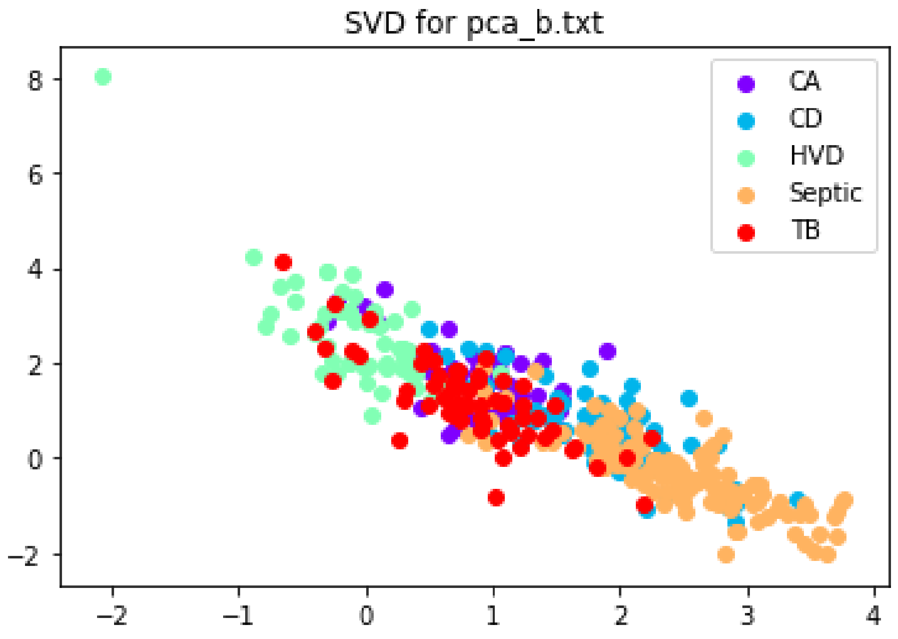
The TruncatedSVD in performed on the adjusted data set X'(X'=X-mean) with no parameters



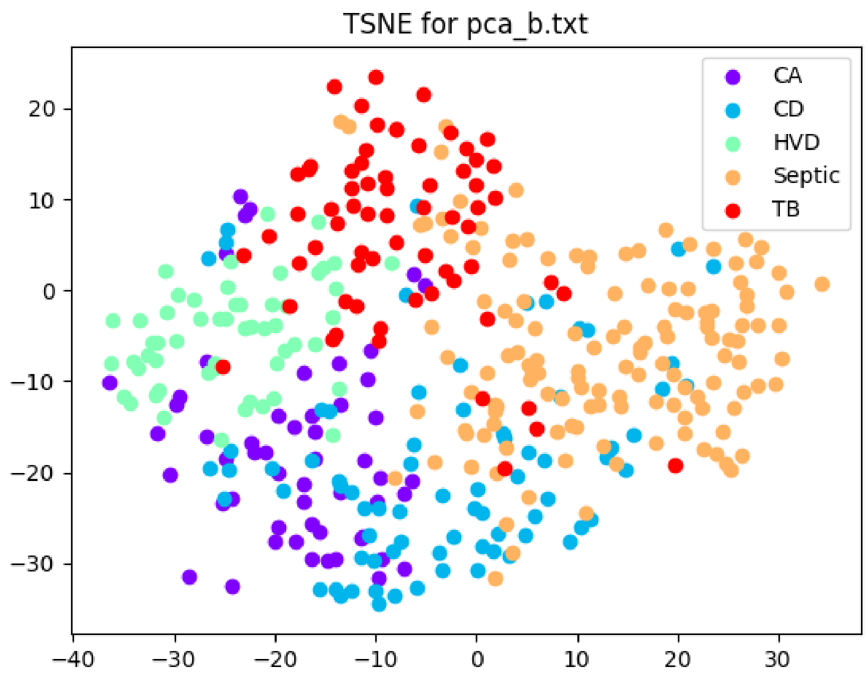
TSNE function is with no parameters

2. pca_b.txt



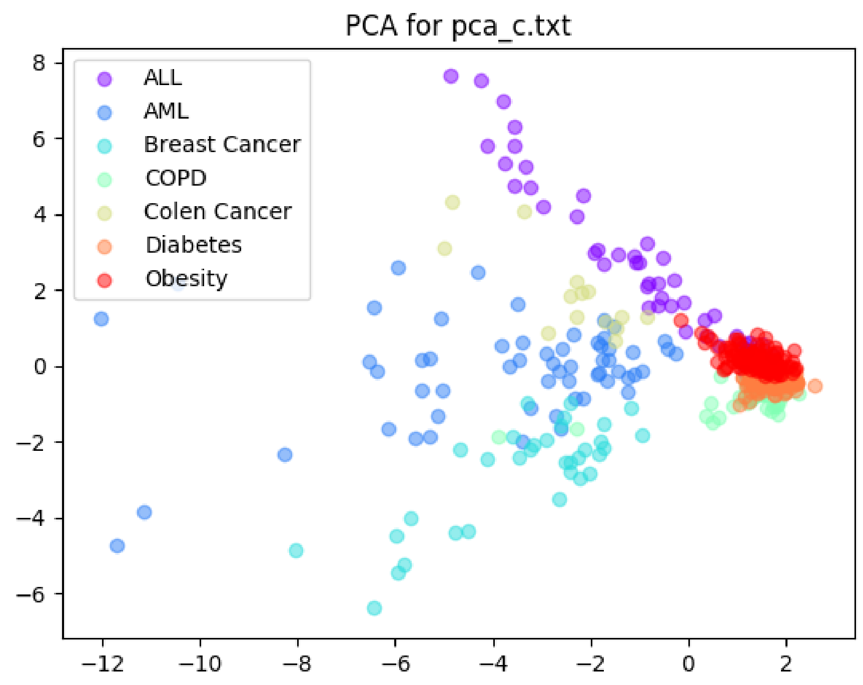


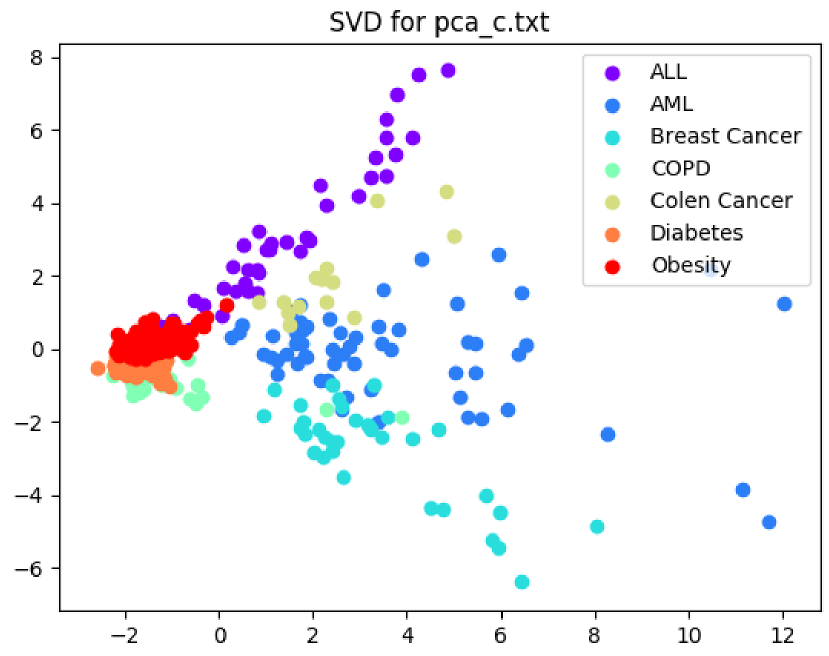
The TruncatedSVD in performed on the adjusted data set X’(X’=X-mean) with no parameters



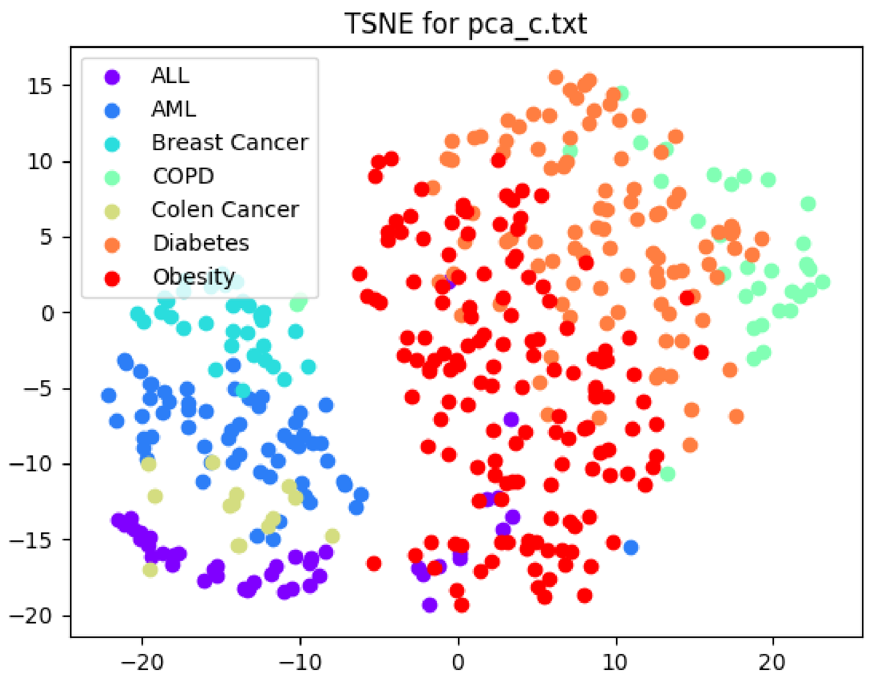
TSNE function is with no parameters

3. pca_c.txt





The TruncatedSVD in performed on the adjusted data set X’(X’=X-mean) with no parameters



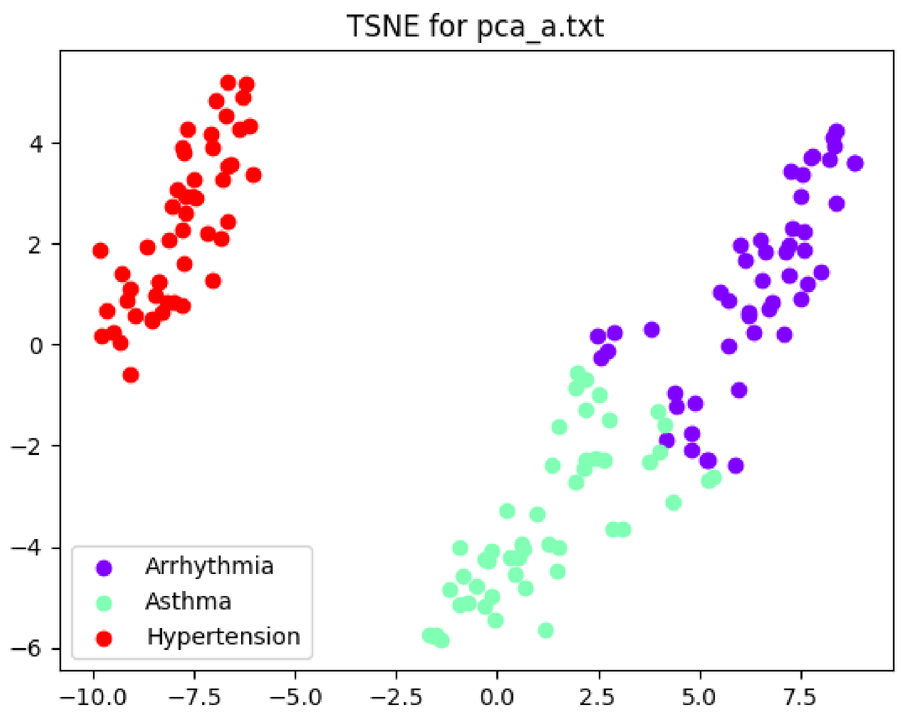
TSNE function is with no parameters

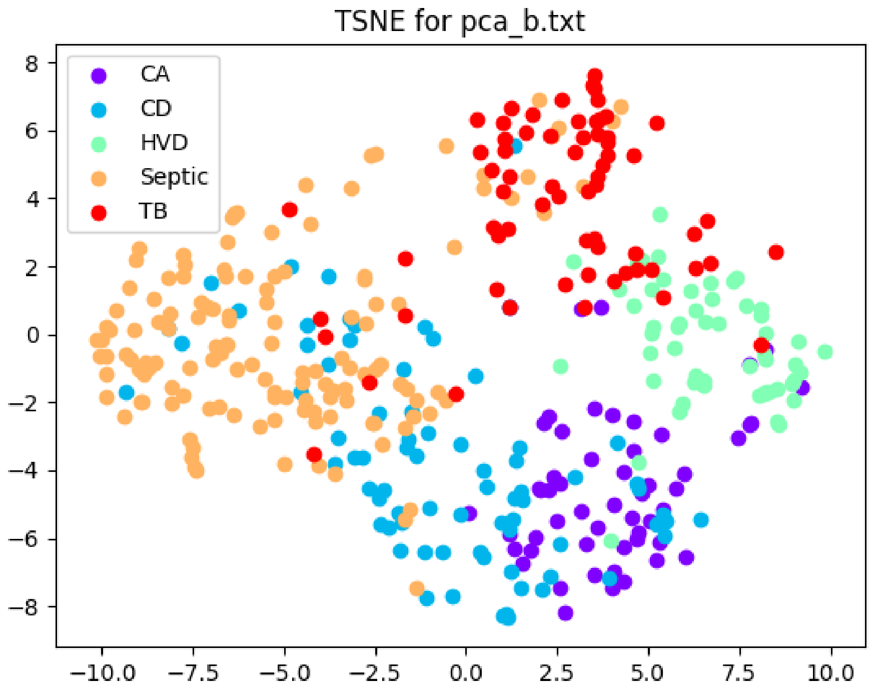
Conclusion: PCA and SVD are mathematical techniques to perform Dimensionality Reduction whereas t-SNE has a probabilistic approach for the same.

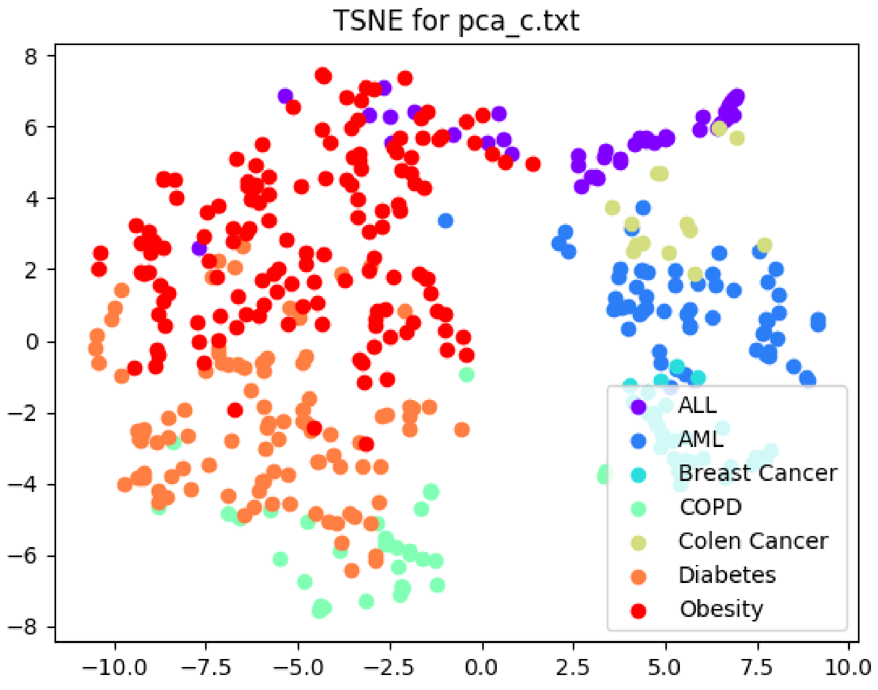
As observed, the output of PCA and SVD is a reduced dimension set with maximum variance. The data is in the form of clusters and is almost same.

The output of t-SNE heavily depends on the parameter values. The proper combination of these parameters may result into plots better than PCA and SVD. In above graphs, we have used a rough t-SNE function which explains the scattered plots.

If we use ***perplexity=30, learning_rate=100, n_iter=1000, init='pca'*** then we get more defined graphs as below:




TSNE for pca_b.txt
CA
CD
HVD
Septic
TB
8
6
4
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


TSNE for pca_c.txt
ALL
AML
Breast Cancer
COPD
Colen Cancer
Diabetes
Obesity
8
6
4
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

References:

1. https://medium.com/towards-data-science/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32
2. http://bigdata-madesimple.com/decoding-dimensionality-reduction-pca-and-svd/
3. https://medium.com/@luckylwk/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b
4. PCA.pptx