# Approximating Principal Component Vectors from Rank-K Approximations for Applications in Genomics

Stanley Yang

## I. Introduction

Over the past decade, the field of genomics has exploded due to plummeting costs of sequencing. In particular, Illumina's short read technology has made sequencing a genome cost from over $100 million to $200 per sequence. Genomics is enabled by innovations in computer science and linear algebra, scaling to TBs of data being generated daily. For example, in single-cell sequencing analysis, statistical machine learning techniques such as PCA and UMAP dimensional reduction is used heavily for clustering analysis.

RNA-seq is a next generation sequencing method that measures the expression of RNA in cells. In RNA-seq specifically, the output gene count matrix is a cell x gene matrix. RNA-seq data is high-dimensional - there are over 20k genes in the genome and experiments can have millions of samples/cells.

Principal component analysis (PCA) is a statistical technique for identifying patterns in high-dimensional data. In genomics, PCA is often used to uncover the underlying structure in large datasets of genetic information (for example, in RNAseq). The principal components of the genomics dataset are the directions along which the data varies the most, and these directions can be approximated using the singular value decomposition (SVD) of the data matrix. However, computing the SVD for large data matrices is computationally intensive. Most scientists using genomics experiments do not have the computational resources to do this quickly. This paper is aimed at using rank-k approximations to efficiently compute the PCA values while retaining most of of variation found using the full SVD.

In this project, I hope to robustly recapitulate the most important principal component vectors for genomics analysis using low rank approximations. I will hope to also explore the tradeoff between accuracy and computational resources.

## II. Dataset

In this project, I used a dataset from a publicly available RNA sequencing experiment. In particular, this contains the RNA sequencing of host response gene expression nasal swabs of COVID patients. This dataset contains a count matrix of cell x gene. The original purpose of this dataset was for time series analysis of patient RNA expression profiles across age, sex, and ethnicity.

The methods for collecting this data and relevant biology experiments are described in the paper "In vivo antiviral host transcriptional response to SARS-CoV-2 by viral load, sex, and age" (linked in citatations).

## III. Methods

In this project, I used two key analysis portions. First, I calculated the Singular Value Decomposition and used that to calculate the low rank approximation of the matrix. Second, I used the PCA values and compared them to the original principal components using cosine similarity.

The Singular Value Decomposition is a key decomposition that can be applied to any m X n matrix dubbed $M$, $M = U\Sigma V^T$. $U$ is a m X m matrix containing the right singular vectors of $M$, $\Sigma$ is the m X n diagonal matrix containing the singular values of $M$, and $V^T$ is a n X n matrix containing the left singular vectors of $M$. $U$ and $V^T$ are orthogonal matrices. This decomposition is important because it is universally applicable to any real matrix, through it is computational intensive.

For low rank approximations of a matrix using the SVD values, Eckert-Young found that the best rank K approximation for a matrix is found by keeping the first $k$ singular values in the $\Sigma$ of the SVD, and zeroing out the rest. This matrix minimizes the matrix norm $\|X - \hat{X}_k\|$. This is more computationally efficient, since there is the zero values in the $\Sigma$ annihilate the last $r-k$ singular values, reducing computation. To calculate $\hat{X}_k$, we approximate it using the equation

$$\sum_{k=1}^{n} \sigma_k u_k v_k^T$$

PCA is a method used for identifying the most important axes of variation. In genomics, this allows us to find the most important sources of genetic variation. PCA is also found using the SVD. First, we demean the values of the matrix (subtracting each row of the matrix by the row sum of said row). Then, we find the principal components by finding the SVD of $\frac{1}{\sqrt{n-1}}M = U\Sigma V^T$, where $M$ is the original matrix that is demeaned. The columns of $U$ are the principal components. These columns describe the direction of the highest variation within the dataset.

Lastly, we compare the principal components using cosine similarity. The cosine similarity describes the angle between the two vectors, such that the angle is minimized at 0 degrees whenever the vectors are the same, and such the angle between two vectors is $0 < x < 180$ degrees. The formula is described as $cos\theta = \frac{x \cdot y}{\|x\|\|y\|}$ Vectors that extremely similar in the same

space will have lower values. This allows us to determine how close our prediction is to our actual PCA components.

## IV. Results

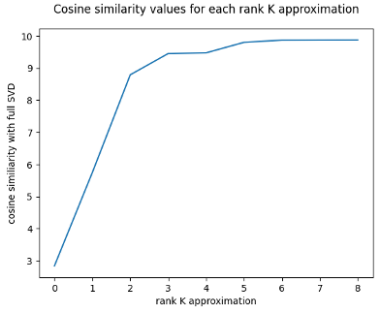


Fig. 1. PC1 approximation using rank K approximation

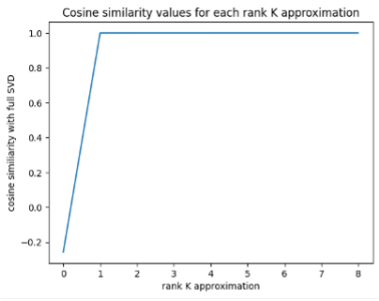


Fig. 2. PC2 approximation using rank K approximation

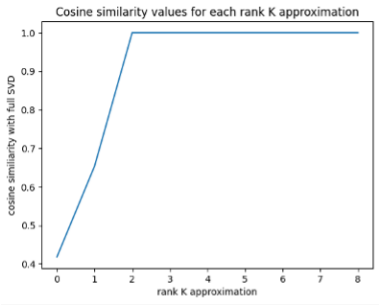


Fig. 3. PC3 approximation using rank K approximation

## V. Results Analysis

First, I graphed the cosine similarity of each original principal component vector to the principal component vectors recovered from the rank K approximations. Essentially, for each principal component vector $ui$ found using PCA, I found the angle similarity between the vectors, essentially showing us a form of distance between the vectors. This allows us to see at what rank k approximation recapitulates what percent of the original principal component vector. For the results, I only used the first three principal components, since a vast majority of the variation should be captured with the earlier singular values since SVD $\Sigma$ is ordered in decreasing order in the diagonal. This means that most of the approximation comes from the earliest $\sigma_i$. Therefore, using a few of the first $\sigma_i$ should have more of an impact on approximating the PCA values than the rest the $\sigma_i$.

I found that most of the principal components can be recovered from a very low rank approximation, around rank 1-3 approximation is sufficient to recover. This seems to vary by principal component, since for the first principal component vector, it takes a rank 4 approximation to reach nearly 100 % similarity with the original PC1. On the other hand, PC2 can be approximated very well using a rank 1 approximation, while PC3 requires a rank 2 approximation to approximate the PC3 well.

## VI. Conclusion

In conclusion, the proposed method for approximating principal component vectors from rank k approximations shows promising results for applications in genomics. I test this on a real world RNA-sequencing experiment, showing that scientists are able to implement this into their current genomic data workflows. I show that it can reduce the computational complexity and hardware requirements of traditional full rank PCA methods, making it more practical for large-scale genomic data analysis. I show that the accuracy is comparable to the current PCA model, meaning that it is viable in current genomics workflows. For researchers working with high dimensional genomic data lacking computational resources, this would be a possible alternative method.

Finding the most important principal components allows genomic scientists to zero down on which genetic variation is most important for identifying different cell types or different gene expression profiles between healthy disease states.

These principal components can be used in order to identify which axis of variation is important to distinguish different disease states, or different cell types in molecular biology. Clinically, it can used to diagnose patients using personalized genomics, where their genomic sequencing data is compared to a diseased patients' genome. In these cases, where thousands of patients are screened and high dimensional data of every patient is used, we need a computational efficient way to find variation in patient groups.

## VII. Future Work/Limitations

For future work, I'd like to be able to test this hypothesis on many other different types of genomics datasets. For example, for single-cell RNA sequencing, a vast majority of the data generated is sparse data, where most of the values are 0. In sparse datasets, knowing the most important few Principal Component vectors is more than enough to find the most of the variation.

I'd also like to redo this project with better hardware capabilities, since computing the rank-K approximations for larger matrices was computationally intensive. This might have been since my method of computing the rank K approximation was less efficient than the optimized np.linalg.svd() function. Another interesting comparison to think about is whether or

not this simple rank-K approximation calculation was faster than the SVD for larger matrices and comparing the difference in times.

## VIII. Acknowledgements

I'd like to thank my family for their constant support and my linear algebra professor Gene Kim for introducing me to the field of linear algebra and its many real world applications.

## IX. Works Cited

1) https://pubmed.ncbi.nlm.nih.gov/32898168/