# PCA and CCA based cell type annotation

Ming Tommy Tang

Divingintogeneticsandgenomics.com

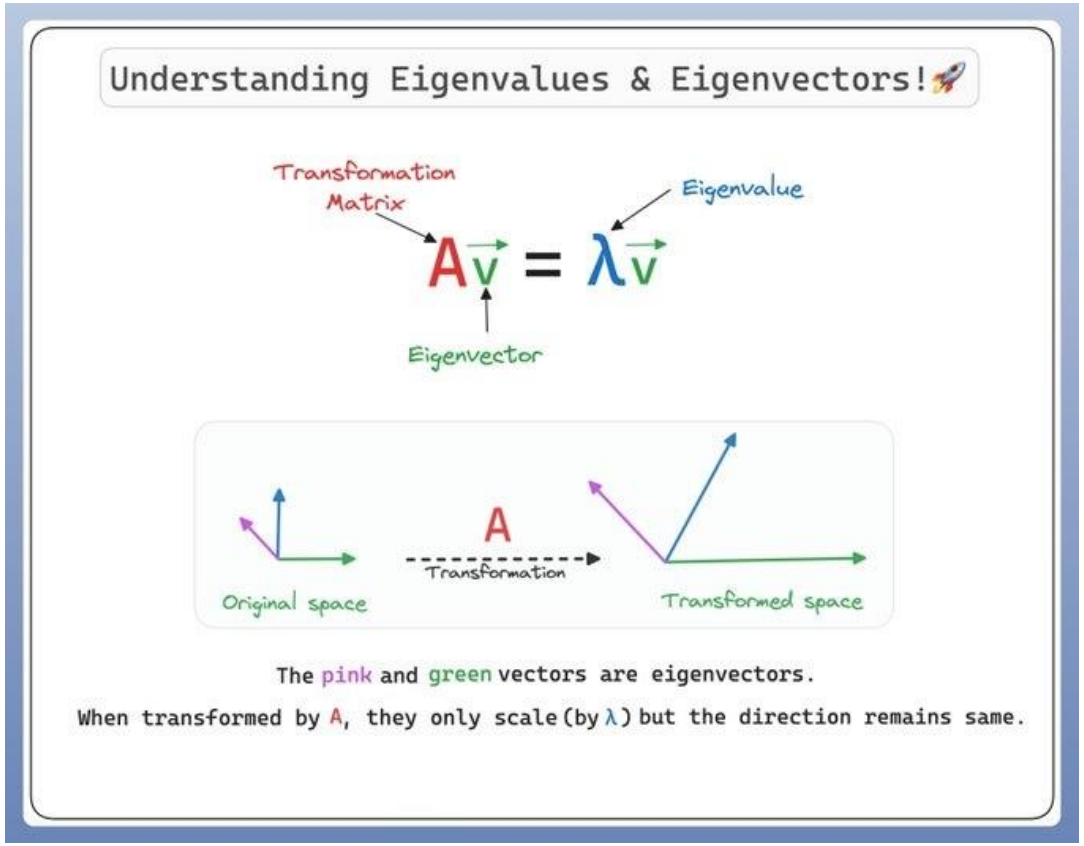
Youtube: Chatomics

04/24/2025

# Cell type annotation (your method is as good as your reference)

- Marker gene based: manual, requires expert biology knowledge, scGate
- correlation-based: singleR, scmap
- Build a multi-class classifier with random forest/XGBoost (or transformers) etc using the reference and predict in test dataset
- Statistical methods: use PCA or CCA (canonical correlation analysis)

# Some fundamentals of eigenvalues and eigenvectors




Watch https://www.3blue1brown.com/lessons/eigenvalues

https://x.com/_avichawla/status/1891374522483548569

# Calculate eigenvalues and eigenvectors

**Finding Eigenvalues and Eigenvectors**

To find the eigenvalues of A, we rearrange the equation:

$$A\vec{v} - \lambda\vec{v} = 0$$

$$(A - \lambda I)\vec{v} = 0$$

Where I is the identity matrix. For this equation to have non-trivial solutions (where $\vec{v} \neq 0$ ), the determinant of $(A - \lambda I)$ must equal zero:

$$\det(A - \lambda I) = 0$$

This is called the characteristic equation. Solving for $\lambda$ gives us the eigenvalues.

For each eigenvalue $\lambda_i$, we can find the corresponding eigenvector $\vec{v}_i$ by solving:

$$(A - \lambda_i I)\vec{v}_i = 0$$

# Suddenly, I understand why I was calculating this in college…

**Example with a 2×2 Matrix**

Let's look at a simple example. Consider the matrix:

$$A = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

To find eigenvalues, we solve:

$$\det \begin{pmatrix} 3-\lambda & 1 \\ 1 & 3-\lambda \end{pmatrix} = 0$$

$$(3-\lambda)^2 - 1 = 0$$

$$(3-\lambda)^2 = 1$$

$$3-\lambda = \pm 1$$

$$\lambda = 2 \text{ or } \lambda = 4$$

So our eigenvalues are 2 and 4.

For $\lambda = 2$, we find eigenvector by solving:

$$(A - 2I)\vec{v} = 0$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

This gives us $v_1 = -v_2$, so an eigenvector is $\vec{v}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ (or any scalar multiple).

For $\lambda = 4$, we get eigenvector $\vec{v}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ (or any scalar multiple).

# Eigen-decomposition: how eigen vectors are useful

### Using Eigenvalues and Eigenvectors

Once we have the eigenvalues and eigenvectors of A, we can perform the eigendecomposition:

$$A = V\Lambda V^{-1}$$

Where:

- $V$ is the matrix whose columns are the eigenvectors of A
- $\Lambda$ is the diagonal matrix of eigenvalues
- $V^{-1}$ is the inverse of V

For our example:

$$V = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}, \Lambda = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$$

### Eigendecomposition Explained

Eigendecomposition is the factorization of a square matrix into a set of eigenvectors and eigenvalues. For a square matrix $A$, the eigendecomposition is:

$$A = V\Lambda V^{-1}$$

Where:

- $V$ is a matrix whose columns are the eigenvectors of $A$
- $\Lambda$ is a diagonal matrix of eigenvalues
- $V^{-1}$ is the inverse of $V$

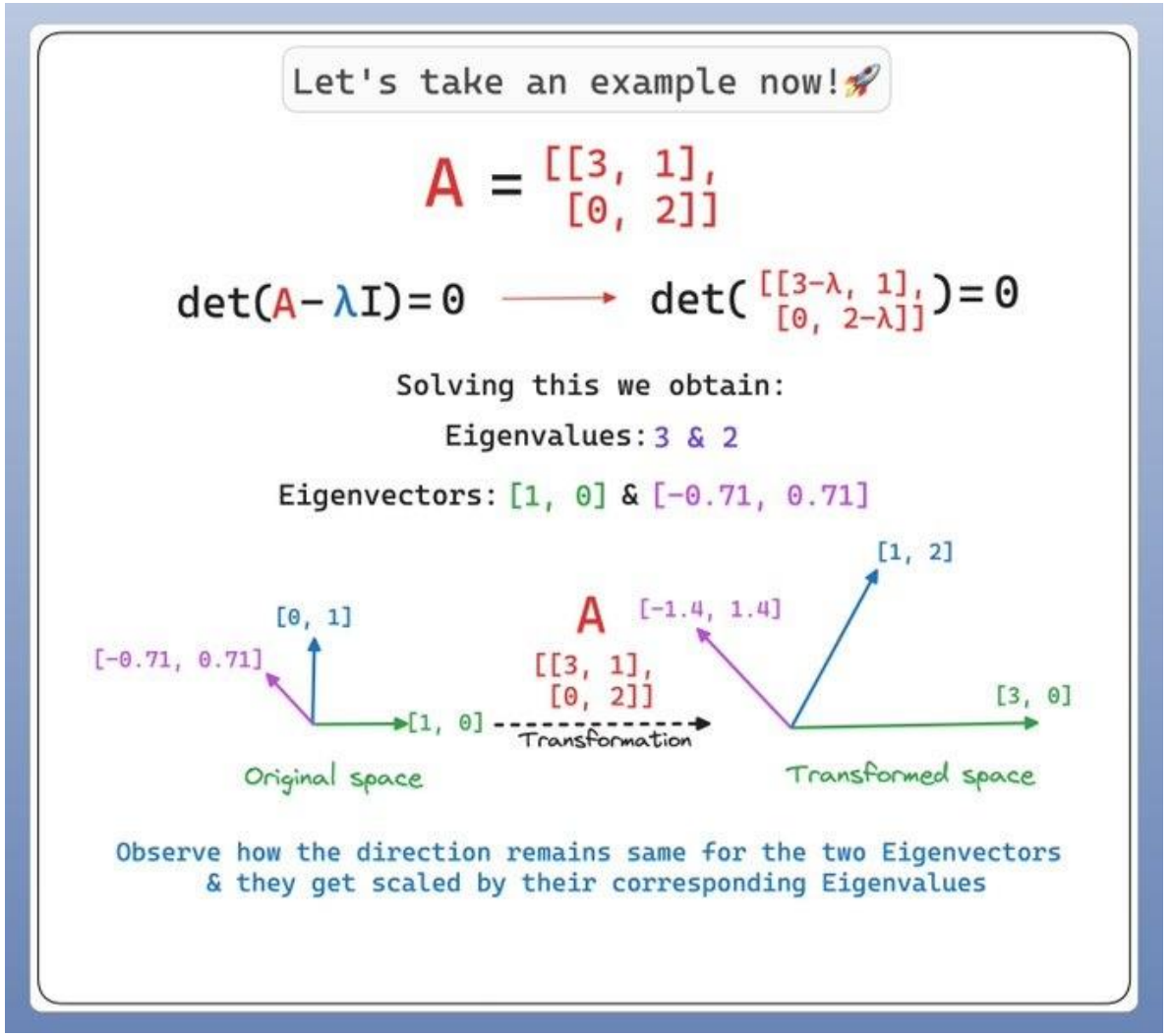
If $A$ is symmetric (like a covariance matrix), then $V$ is orthogonal, so $V^{-1} = V^T$, giving us:

$$A = V\Lambda V^T$$

# What does it mean geometrically timing a vector with a matrix A?

A %*% B

Linear combination of the columns of matrix A from columns of B



# What is PCA?

## Connection to PCA

Principal Component Analysis (PCA) is fundamentally an eigenvalue problem. Here's the connection:

1. In PCA, we start with a data matrix $X$ (with rows as observations and columns as features)
2. We compute the covariance matrix $\Sigma = \frac{1}{n-1} X^T X$ (assuming $X$ is centered)
3. The eigenvectors of this covariance matrix are the principal components (directions of maximum variance)
4. The corresponding eigenvalues tell us how much variance is explained by each principal component

# Singular value decomposition (SVD) and eigendecomposition

PCA can be computed using either eigendecomposition or Singular Value Decomposition (SVD):

**Eigendecomposition approach:**

We decompose the covariance matrix $\Sigma$ as:

$$\Sigma = V\Lambda V^T$$

Where:

- $V$ is the matrix of eigenvectors (principal components)
- $\Lambda$ is a diagonal matrix of eigenvalues (variances along principal components)

**SVD approach:**

We decompose the centered data matrix $X$ directly:

$$X = UDV^T$$

Where:

- $U$ contains the left singular vectors
- $D$ is a diagonal matrix of singular values
- $V$ contains the right singular vectors (same as eigenvectors of $X^T X$)

**Connection to PCA**

Principal Component Analysis (PCA) is fundamentally an eigenvalue problem. Here's the connection:

1. In PCA, we start with a data matrix $X$ (with rows as observations and columns as features)
2. We compute the covariance matrix $\Sigma = \frac{1}{n-1} X^T X$ (assuming $X$ is centered)
3. The eigenvectors of this covariance matrix are the principal components (directions of maximum variance)
4. The corresponding eigenvalues tell us how much variance is explained by each principal component

**The connection:**

- The right singular vectors in $V$ from SVD are identical to the eigenvectors of the covariance matrix
- The singular values in $D$ relate to the eigenvalues by: $\lambda_i = \frac{d_i^2}{n-1}$
- $UD$ gives us the principal components scores (data projected onto principal components)

This means when we use SVD for PCA, the $V$ matrix gives us the principal directions (eigenvectors of the covariance matrix), the squares of the singular values (divided by $n-1$) give us the eigenvalues (variances), and $UD$ gives us the data projected onto those directions.

# Mathematically demonstrate SVD on the matrix of X and eigendecomposition on covariance matrix of X are the same

**The Proof of Equivalence**

Let's show these approaches are mathematically equivalent:

Starting with the SVD: $X = UDV^T$

The covariance matrix $\Sigma$ can be written as:

$$\Sigma = \frac{1}{n-1}X^TX = \frac{1}{n-1}(UDV^T)^T(UDV^T)$$

Simplifying:

$$\Sigma = \frac{1}{n-1}VDU^TUDV^T$$

Since $U$ has orthonormal columns, $U^TU = I$:

$$\Sigma = \frac{1}{n-1}VD^2V^T$$

This is an eigendecomposition of $\Sigma$ where:

- $V$ is the matrix of eigenvectors
- $\frac{1}{n-1}D^2$ is the diagonal matrix of eigenvalues $\Lambda$

Therefore:

$$\Sigma = V\left(\frac{1}{n-1}D^2\right)V^T = V\Lambda V^T$$

This shows that the eigenvectors of $\Sigma$ (used in PCA via eigendecomposition) are exactly the right singular vectors from SVD, and the eigenvalues are related to the singular values by:

$$\lambda_i = \frac{d_i^2}{n-1}$$

**Equivalence of Scores**

For the principal component scores:

Using eigendecomposition: $\text{Scores} = XV$

Using SVD: $\text{Scores} = UD$

Let's show these are equivalent:

$$XV = UDV^TV = UD$$

The last step follows because $V^TV = I$ since $V$ has orthonormal columns.

Therefore, the principal component scores are identical whether calculated through eigendecomposition or SVD.

Rows are samples,

Columns are PC1..PC2...PCn

When we say project Y to the PCA space of X: we do SVD of X to get the Vx

And then Y %*% Vx

# In R, the prcomp() function uses svd() internally

r-source / src / library / stats / R / **prcomp.R**

Code | Blame | 148 lines (139 loc) · 4.84 KB

```
19  prcomp <- function (x, ...) UseMethod("prcomp")
20
21  prcomp.default <-
22      function(x, retx = TRUE, center = TRUE, scale. = FALSE, tol = NULL, ...)
23  {
24      chkDots(...)
25      x <- as.matrix(x)
26      x <- scale(x, center = center, scale = scale.)
27      cen <- attr(x, "scaled:center")
28      sc <- attr(x, "scaled:scale")
29      if(any(sc == 0))
30          stop("cannot rescale a constant/zero column to unit variance")
31      s <- svd(x, nu = 0)
32      s$d <- s$d / sqrt(max(1, nrow(x) - 1))
33      if (!is.null(tol)) {
34          ## we get rank at least one even for a 0 matrix.
35          rank <- sum(s$d > (s$d[1L]*tol))
36          if (rank < ncol(x)) {
37              s$v <- s$v[, 1L:rank, drop = FALSE]
38              s$d <- s$d[1L:rank]
39          }
40      }
```

https://github.com/SurajGupta/r-source/blob/master/src/library/stats/R/prcomp.R#L31

# Canonical Correlation Analysis (CCA) and

**Formulas for CCA**

Input Data:

a matrix $X \in \mathbb{R}^{p \times n_1}$ is the matrix of gene expression from the first dataset (3k cells), and $Y \in \mathbb{R}^{p \times n_2}$ is the matrix of gene expression from the second dataset (10k cells).

**Canonical Variates**: We aim to find vectors $a \in \mathbb{R}^p$ and $b \in \mathbb{R}^p$ such that:

$$u = X^T a \quad (\text{canonical variate for cells in } X)$$

$$v = Y^T b \quad (\text{canonical variate for cells in } Y)$$

Where $u$ and $v$ are the canonical variates, and the goal is to maximize the correlation:

$$\text{corr}(u, v) = \frac{\text{cov}(u, v)}{\sqrt{\text{var}(u)} \cdot \sqrt{\text{var}(v)}}$$

**Covariance Matrices**:

$$\Sigma_{XX} = \frac{1}{n_1 - 1} XX^T \quad (\text{covariance matrix of dataset } X)$$

$$\Sigma_{YY} = \frac{1}{n_2 - 1} YY^T \quad (\text{covariance matrix of dataset } Y)$$

$$\Sigma_{XY} = \frac{1}{\min(n_1, n_2) - 1} X^T Y \quad (\text{cross-covariance matrix between } X \text{ and } Y)$$

**Generalized Eigenvalue Problem (GEP)**:

To find the canonical weights $a$ and $b$, we solve the generalized eigenvalue problem:

$$\Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} a = \lambda a$$

Similarly for $b$:

$$\Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} b = \lambda b$$

The eigenvalue $\lambda$ corresponds to the canonical correlation. $a$ is the canonical weight vector for dataset $X$ $b$ is the canonical weight vector for dataset $Y$

Solving the eigenvalue problem involves finding the eigenvalues and eigenvectors that maximize the correlation between the datasets, but it effectively solves the same problem as the `SVD` for the covariance matrix $\Sigma XY$. One can

Reference: https://divingintogeneticsandgenomics.com/post/cca-alignment/

# How SVD is used and how it relates to the GEP.

**Perform SVD on the Cross-Covariance Matrix**:

The SVD of the cross-covariance matrix $\Sigma_{XY}$ can be expressed as:

$$\Sigma_{XY} = UDV^T$$

For U: rows are cells from X, columns are CCA1,CCA2..
For V: rows are cells from Y, columns are CCA1, CCA2..

where:

- $U$ is a matrix of left singular vectors (canonical directions for dataset $X$,
- $V$ is a matrix of right singular vectors (canonical directions for dataset $Y$,
- $D$ is a diagonal matrix containing singular values, which represent the strength of correlations.

The matrix $U$ contains the left singular vectors of $\Sigma_{XY}$, which correspond to the canonical directions for dataset $X$. These directions are the axes along which the data in $X$ is maximally correlated with the data in $Y$, as measured by the singular values in $D$. Thus, $U$ represents the canonical variates (directions) for $X$, just as $V$ represents the canonical variates for $Y$.

https://xinmingtu.cn/blog/2022/CCA_dual_PCA/

# In summary

- PCA: SVD on the matrix X
- CCA: SVD on the cross-covariance matrix XY

# Mathematically demonstrate SVD on the matrix of X and eigendecomposition on covariance matrix of X are the same

### The Proof of Equivalence

Let's show these approaches are mathematically equivalent:

Starting with the SVD: $X = UDV^T$

The covariance matrix $\Sigma$ can be written as:

$$\Sigma = \frac{1}{n-1}X^TX = \frac{1}{n-1}(UDV^T)^T(UDV^T)$$

Simplifying:

$$\Sigma = \frac{1}{n-1}VDU^TUDV^T$$

Since $U$ has orthonormal columns, $U^TU = I$:

$$\Sigma = \frac{1}{n-1}VD^2V^T$$

This is an eigendecomposition of $\Sigma$ where:

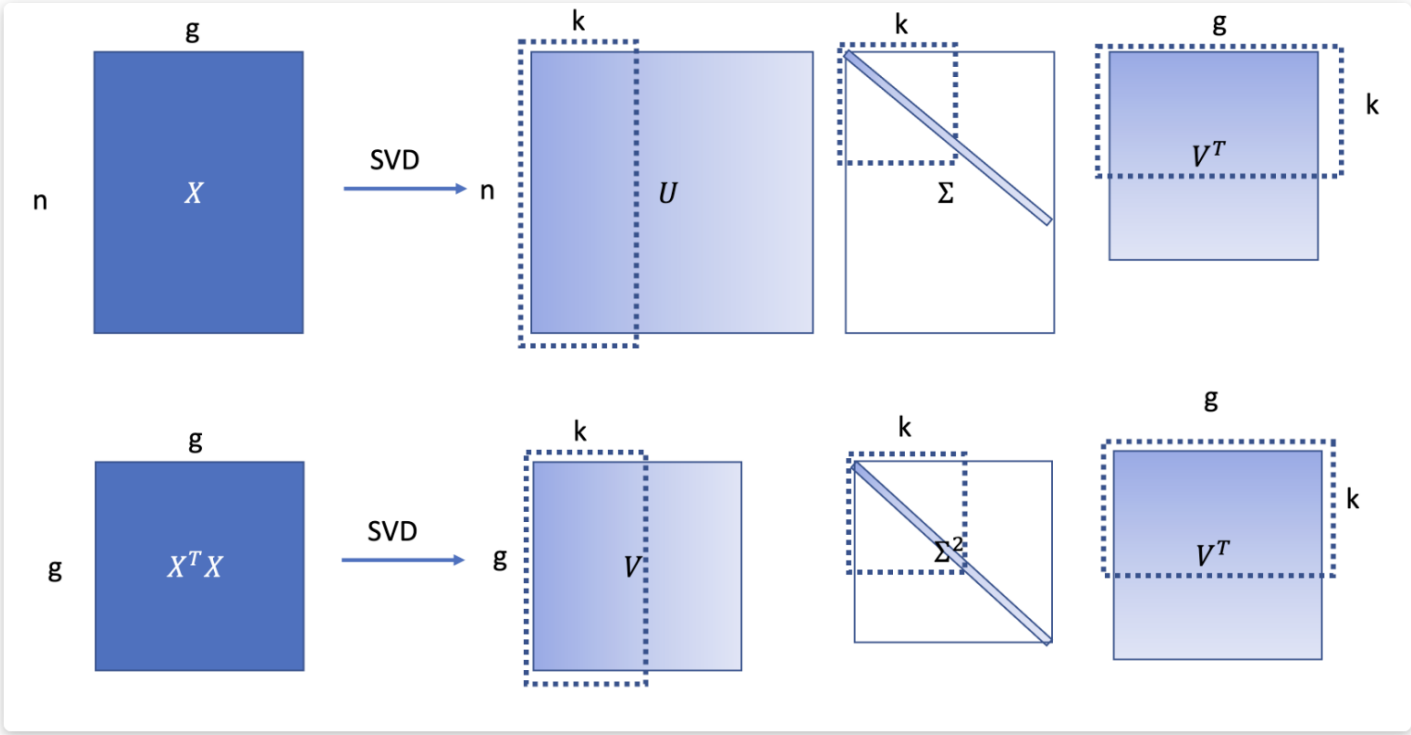
- $V$ is the matrix of eigenvectors
- $\frac{1}{n-1}D^2$ is the diagonal matrix of eigenvalues $\Lambda$

Therefore:

$$\Sigma = V\left(\frac{1}{n-1}D^2\right)V^T = V\Lambda V^T$$

The solution $V$ can be calculated from the singular value decomposition (SVD) of $X$:

$$X = U\Sigma V^T, X^TX = V\Sigma^2V^T.$$

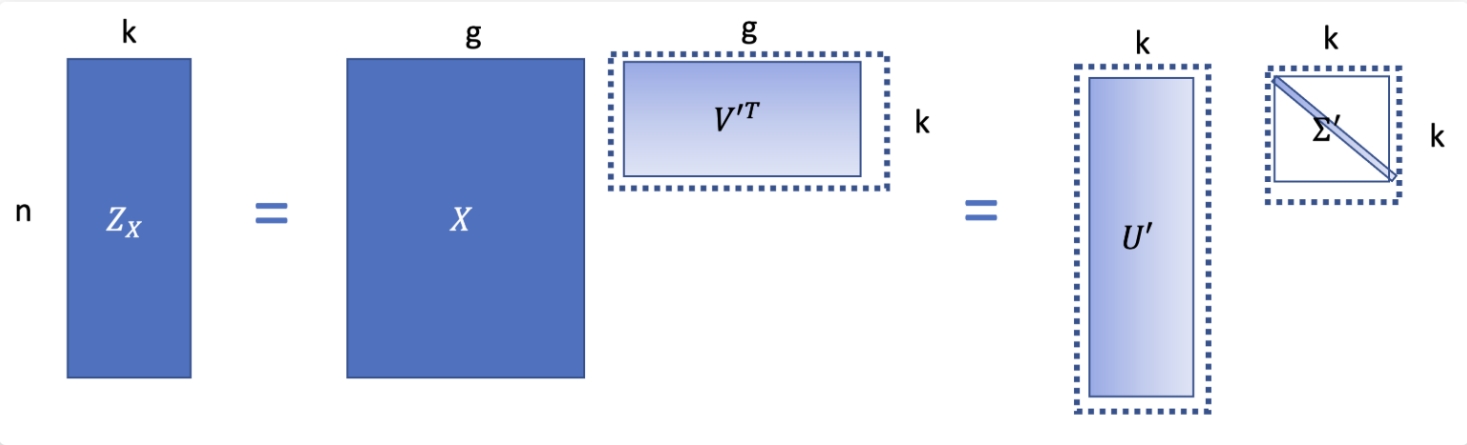



$X \in R^{n\times g}$ is the gene-gene expression matrix. $X^TX \in R^{g\times g}$ are the gene co-variance matrix. Equivalently put, the SVD provides the best low rank approximation of gene-gene co-variance matrix as the multiplication of top $K$ singular values and vectors of $X$ shown below:

https://xinmingtu.cn/blog/2022/CCA_dual_PCA/

And the embedding we are looking for in PCA will be

$$Z = XV_{1:g,1:k} = U\Sigma V^T V_{1:g,1:k} = U_{1:n,1:k}\Sigma_{1:k}.$$




$Z_X \in R^{n\times k}$ is the low dimensional embeddings. } $TheV'^T \in R^{g\times k}$ is the project matrix.Then based on the SVD, $Z = U_{1:n,1:k}\Sigma_{1:k}$

## Equivalence of Scores

For the principal component scores:

Using eigendecomposition: $\text{Scores} = XV$

Using SVD: $\text{Scores} = UD$

Let's show these are equivalent:

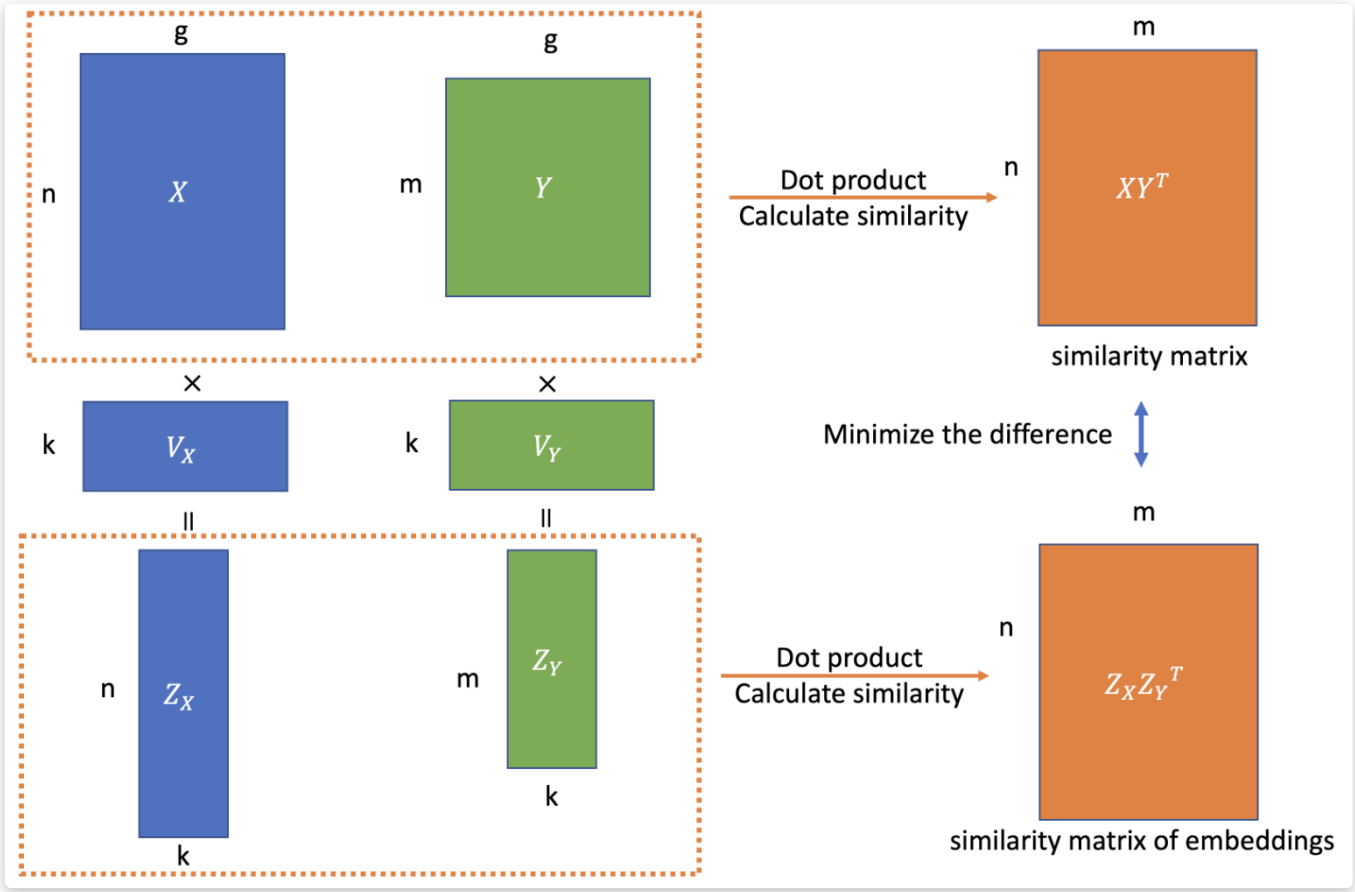
$$XV = UDV^TV = UD$$

The last step follows because $V^TV = I$ since $V$ has orthonormal columns.

Therefore, the principal component scores are identical whether calculated through eigendecomposition or SVD.

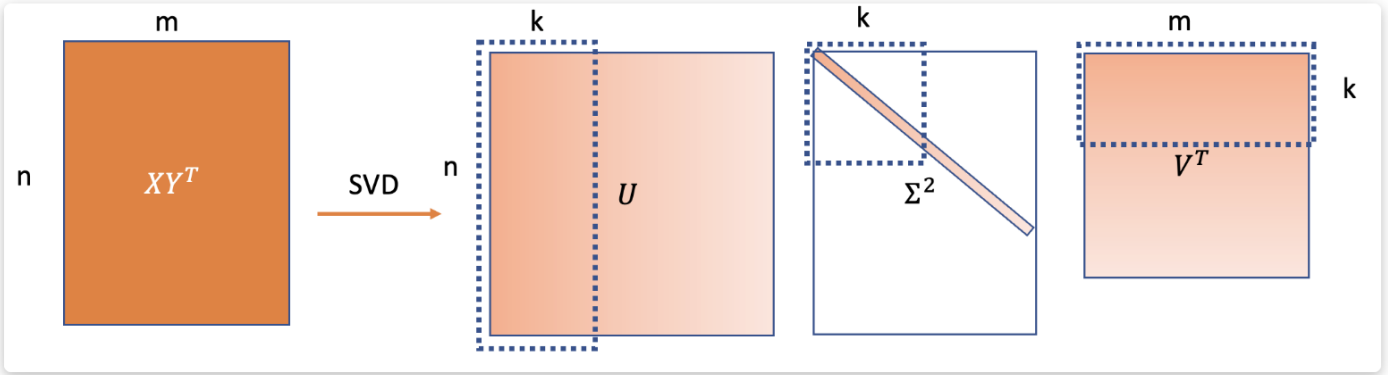
Gene loadings

Cell loadings



# SVD on the cross-covariance matrix XY^T

$$XY^T = U\Sigma V^T$$

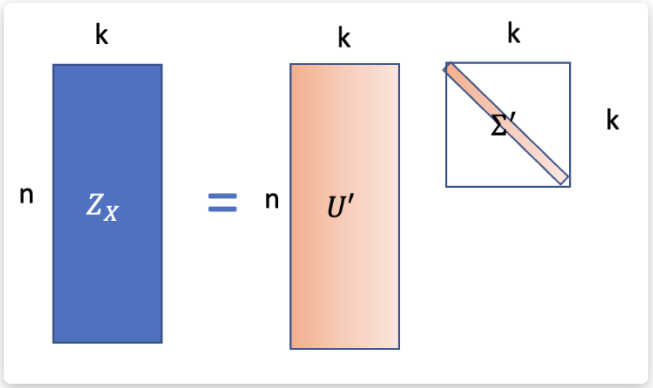


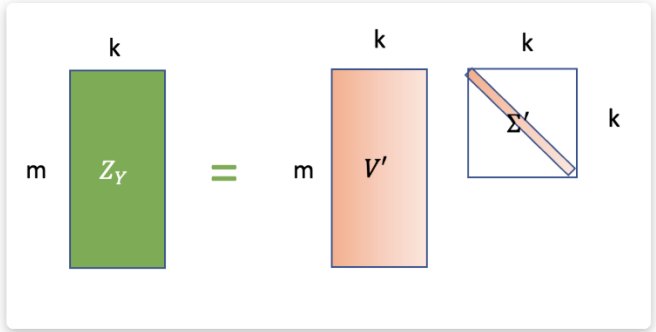

XY^T = UD * DV^T

Zx=UD
Zy=VD

The best embeddings will be

$$Z_X = U_{1:n,1:k}(\Sigma_{1:k})^{\frac{1}{2}}, Z_Y = V_{1:n,1:k}(\Sigma_{1:k})^{\frac{1}{2}}$$

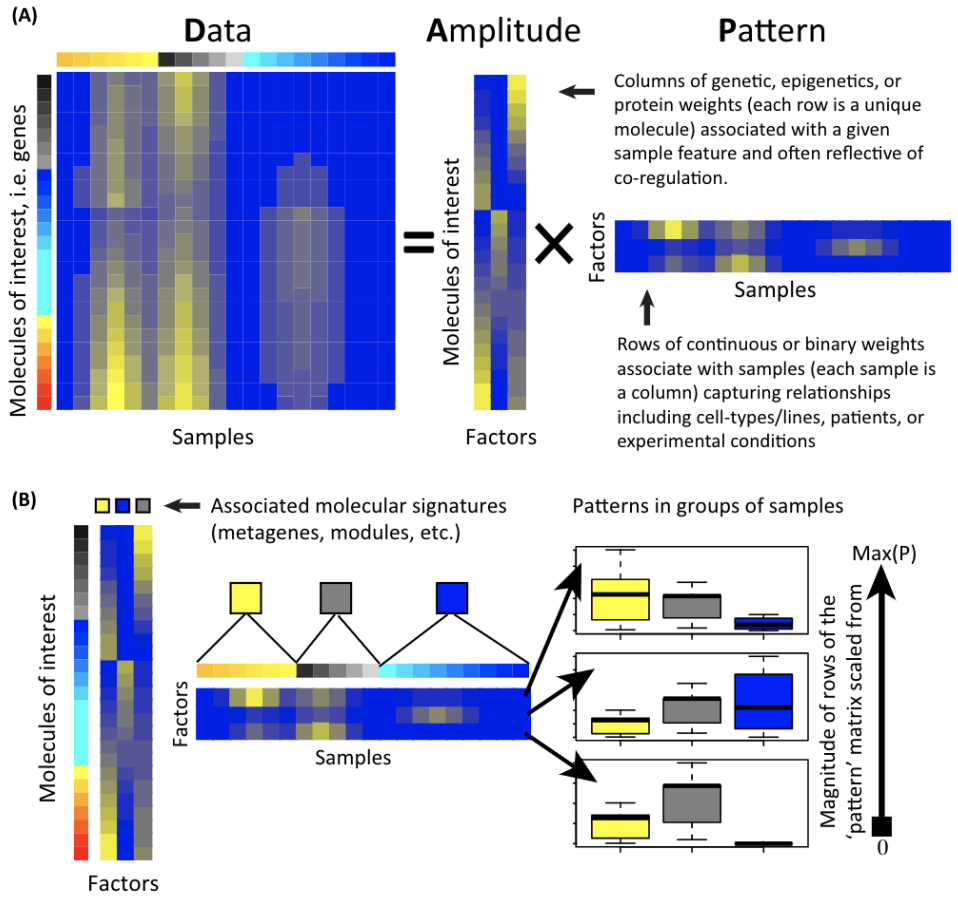



$Z_X \in R^{n\times k}, Z_Y \in R^{m\times k}$ are the best low dimensional embeddings(projection) of two batch data $X, Y$.

https://xinmingtu.cn/blog/2022/CCA_dual_PCA/

# Matrix Factorization for biological data




PCA: X=UDV^T
The Amplitude matrix: is V gene loading matrix
The Pattern matrix: is the UD or XV: cell/sample loading

NMF (non-negative matrix factorization: X = W %*% H
ICA (independent component analysis): X= S %*% A

https://www.sciencedirect.com/science/article/pii/S0168952518301240

https://divingintogeneticsandgenomics.com/post/matrix-factorization-for-single-cell-rnaseq-data/