EDA: Principal Component Analysis (PCA)

RNA-seq data analysis

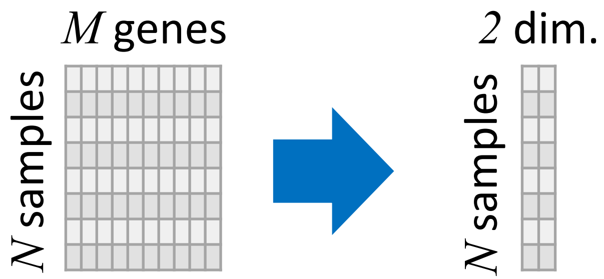
Paulo Czarnewski

https://czarnewski.github.io/czarnewski/index.html

Why PCA?

Simplify complexity, so it becomes easier to work with.

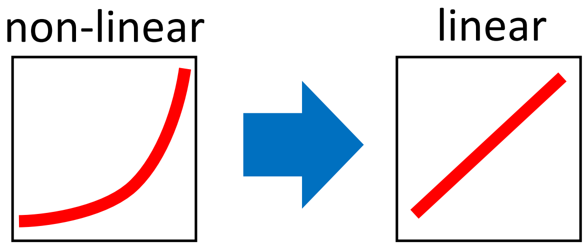
- Reduce number of features (genes)
- Need to transform non-linear relationships to linear



“Remove” redundancies in the data

Identify the most relevant information

Find and filter noise



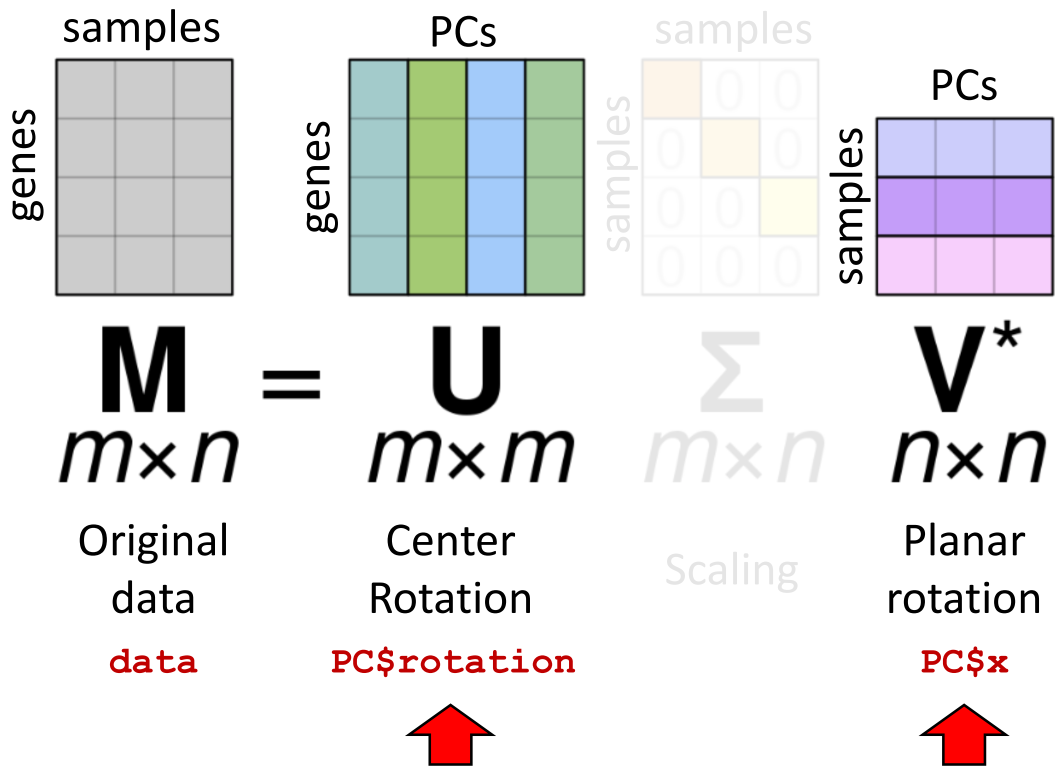
Data visualization

How PCA works

It is a LINEAR algebraic method of dimensionality reduction.

It is a case inside Singular Value Decomposition (SVD) method (data compression)

Any matrix can be decomposed as a multiplication of other matrices (Matrix Factorization).



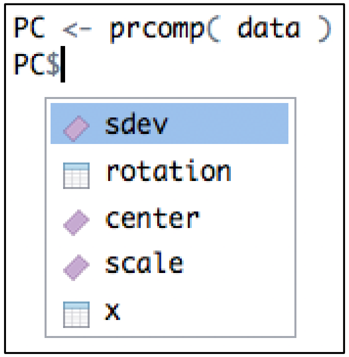
$$\underset{m \times n}{\mathbf{M}} = \underset{m \times m}{\mathbf{U}} \; \underset{m \times n}{\Sigma} \; \underset{n \times n}{\mathbf{V}^*}$$

Original data — `data`

Center Rotation — `PC$rotation`

Scaling

Planar rotation — `PC$x`

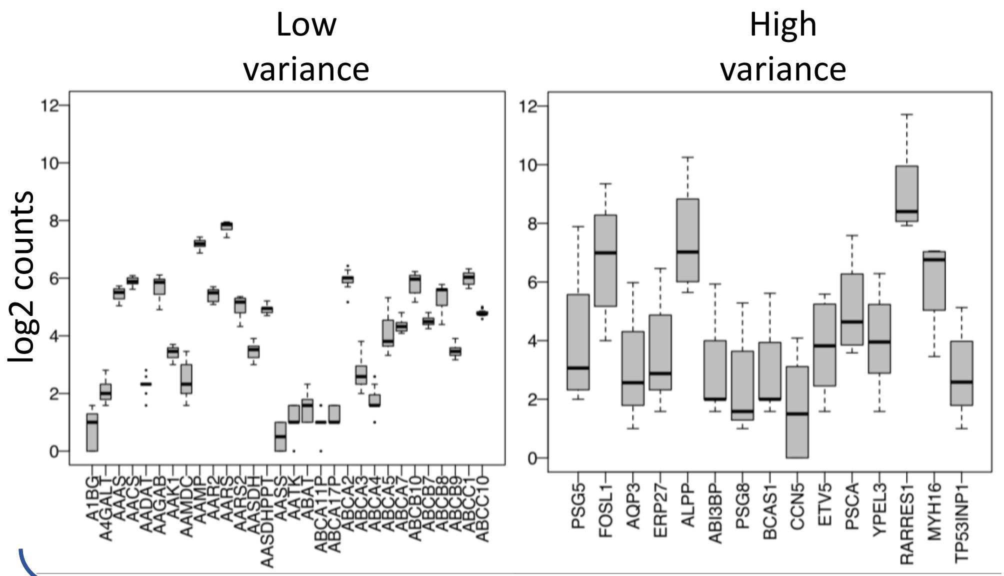


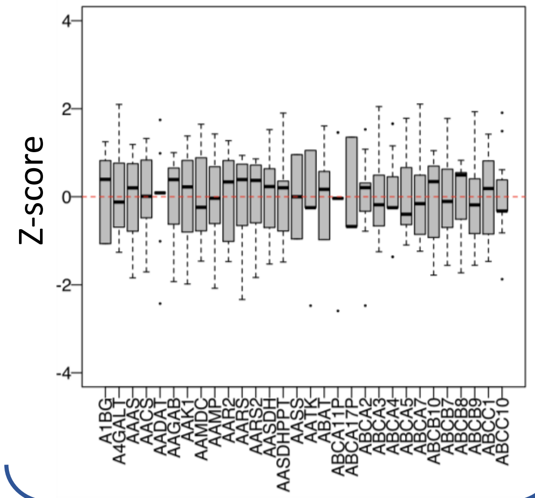
```
PC <- prcomp( data )
PC$
```

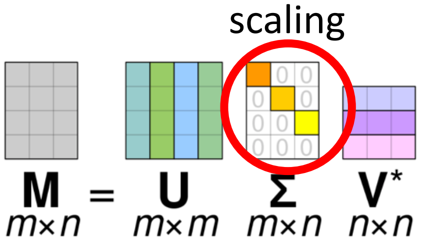
Data transformation and scaling

Before applying PCA, the data should be first transformed to a linear scale (i.e. log)

Each feature should be scaled to have a similar center (zero) and similar deviation.





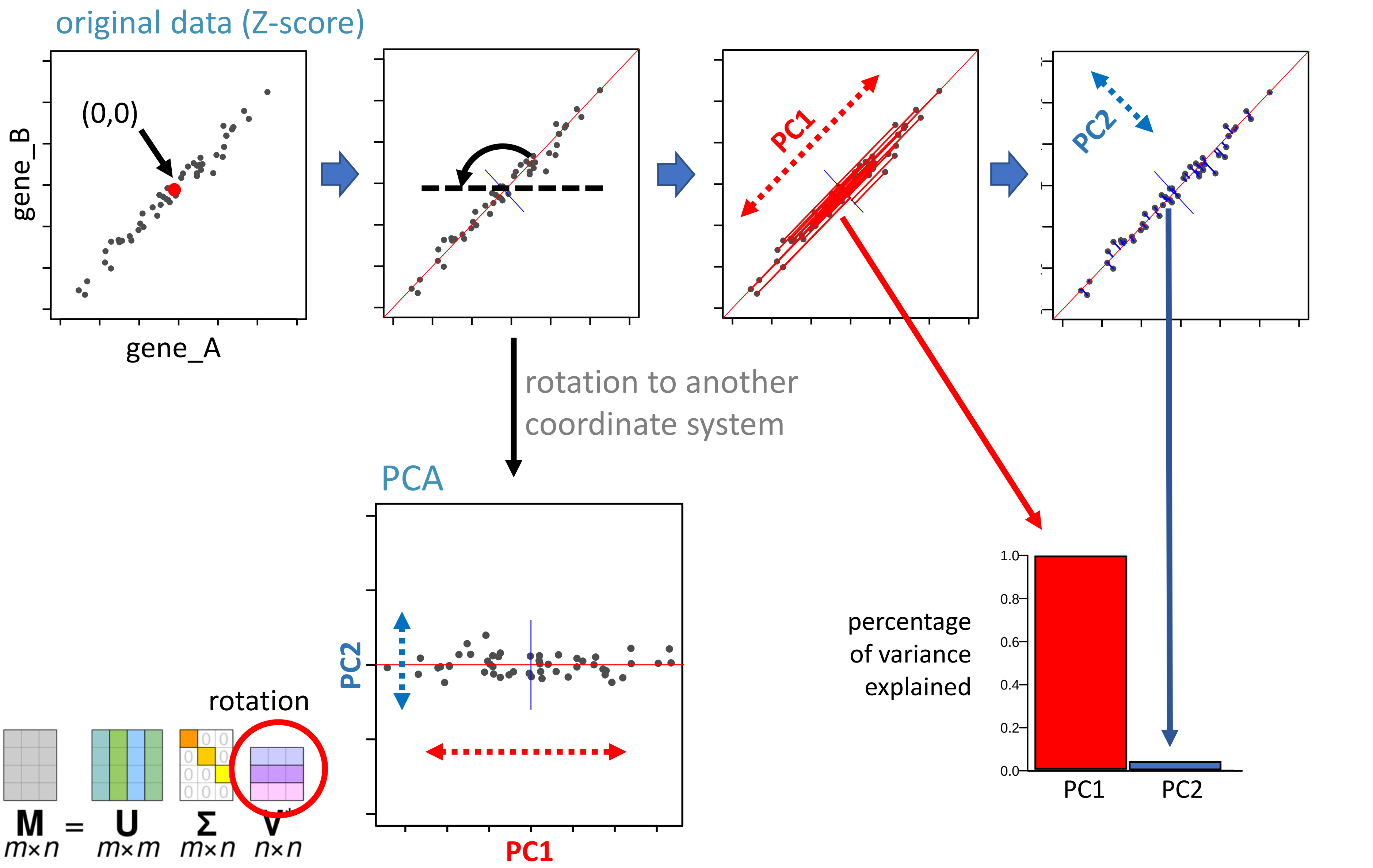


PCA on raw counts will separate genes with higher counts in the first PCs

(higher distance to 0)

PCA on Z-score will separate genes with most common expression trends in the first PCs

How PCA works



original data (Z-score)

rotation to another coordinate system

PCA

percentage of variance explained

rotation

$$\underset{m\times n}{\mathbf{M}} = \underset{m\times m}{\mathbf{U}}\ \underset{m\times n}{\mathbf{\Sigma}}\ \underset{n\times n}{\mathbf{V}^t}$$

How PCA works

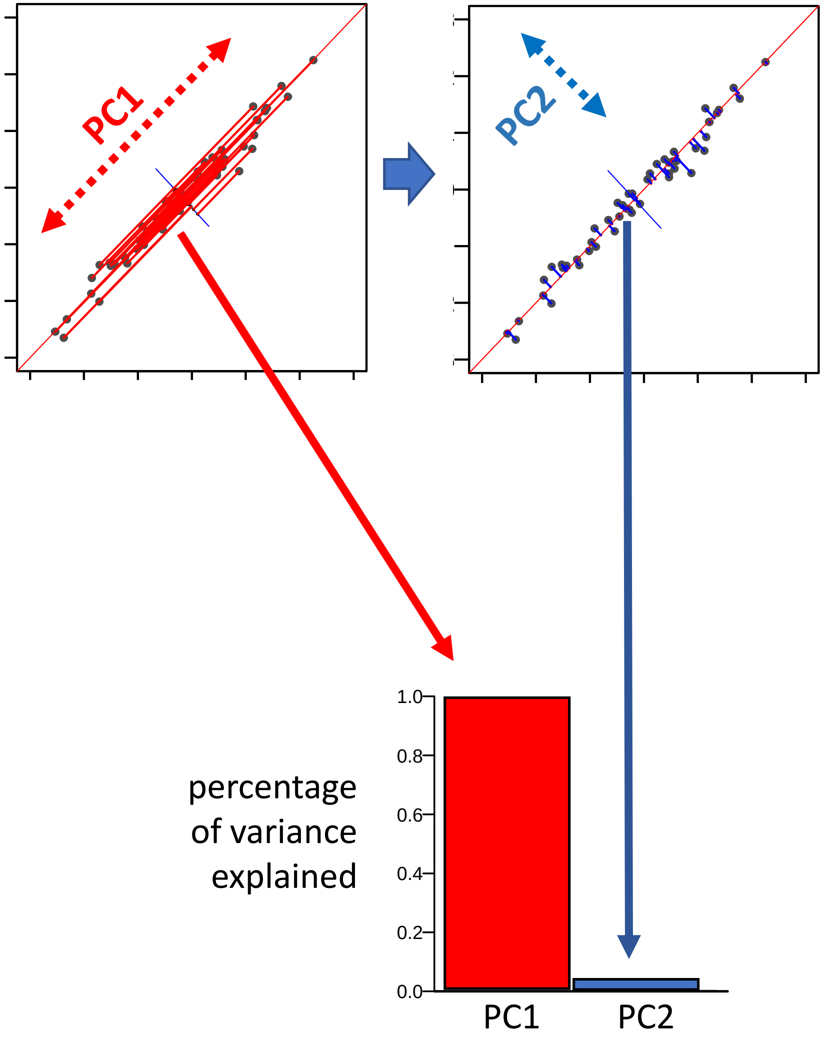
PC1 explains >98% of the variance

1 PC thus represents 2 genes very well

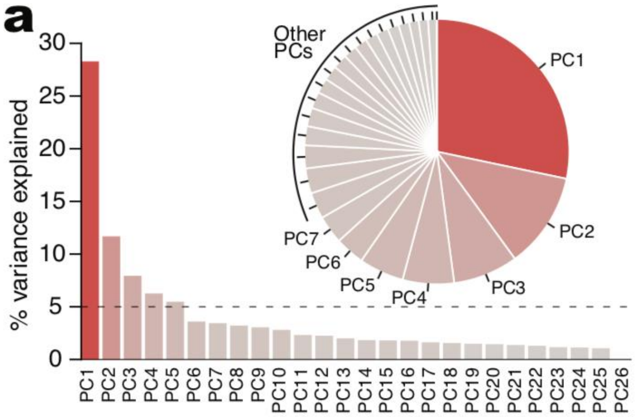
“Removing” redundancy

PC2 is nearly insignificant in this example

Could be disregarded



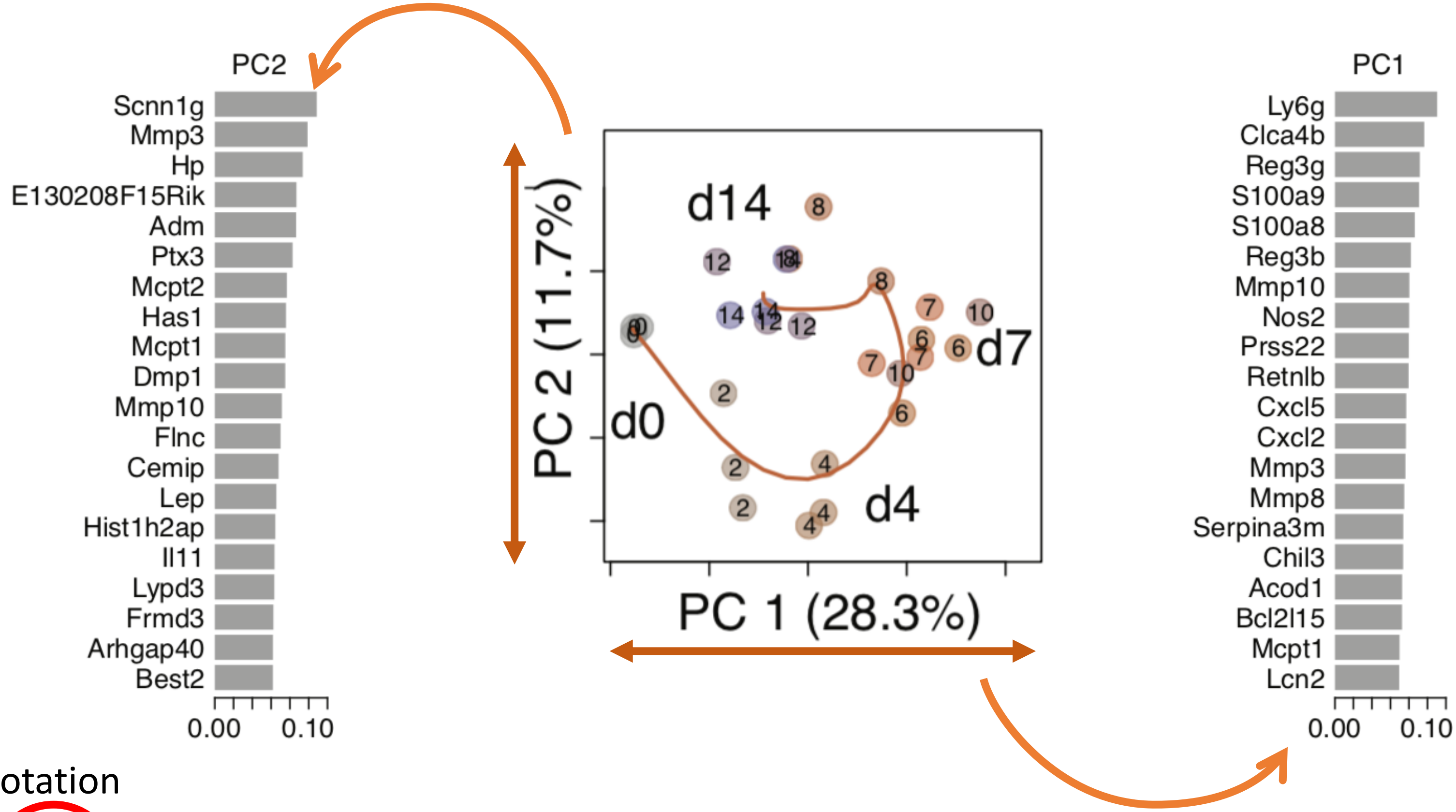
In real life …

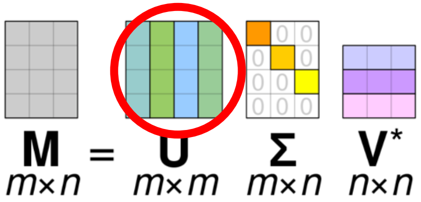


Czarnewski *et al* 2019

How PCA works

Each PC has a meaning



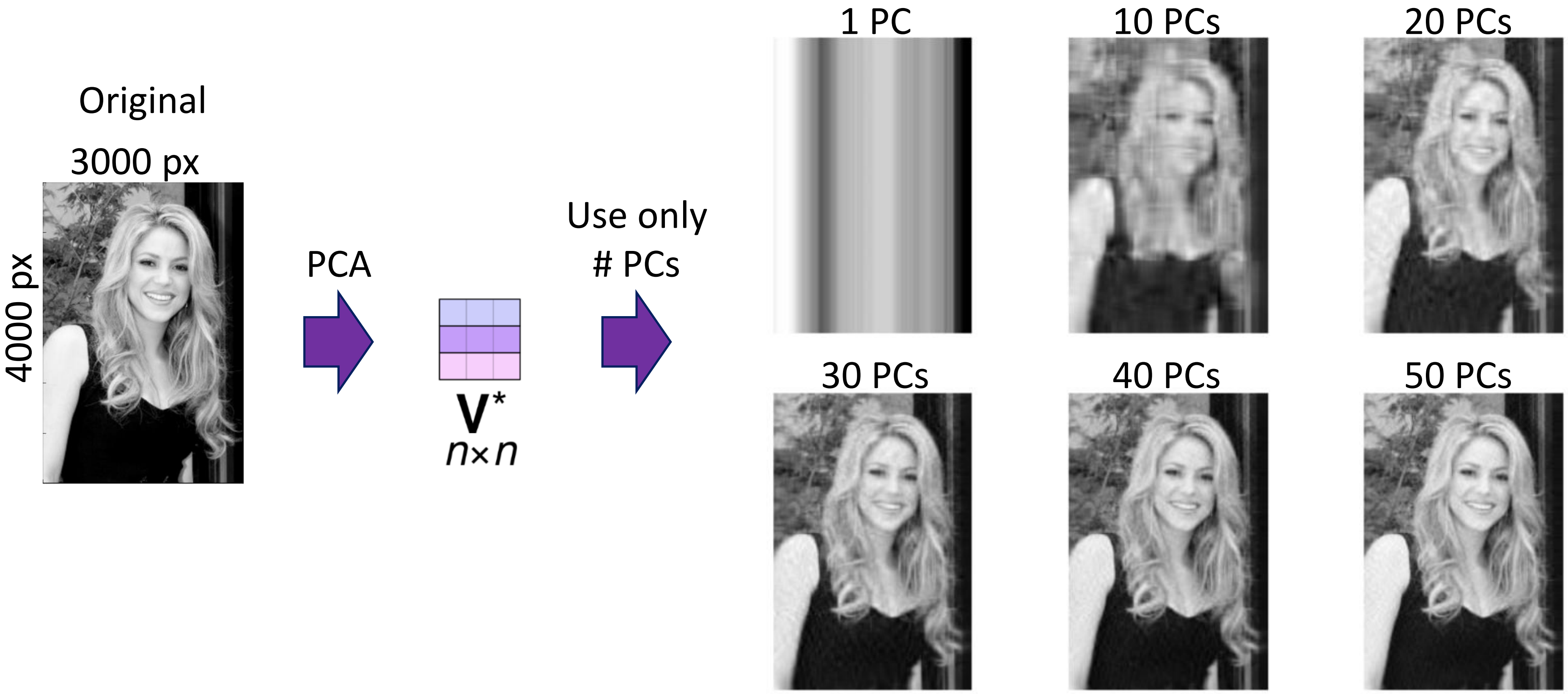


$$\underset{m\times n}{\mathbf{M}} = \underset{m\times m}{\mathbf{U}}\ \underset{m\times n}{\mathbf{\Sigma}}\ \underset{n\times n}{\mathbf{V}^*}$$

Czarnewski *et al* 2019

A visual intuition of PCA

The top principal components store more important ~~Shakira~~ information



Original

3000 px

4000 px

PCA

$\mathbf{V}^{*}_{n\times n}$

Use only
# PCs

1 PC

10 PCs

20 PCs

30 PCs

40 PCs

50 PCs

https://glowingpython.blogspot.com/2011/07/pca-and-image-compression-with-numpy.html

PCA summary

It is a LINEAR method of dimensionality reduction

The data is usually SCALED (i.e. Z-score) and TRANSFORMED (i.e. log) prior to PCA

It is an interpretable dimensionality reduction

The top principal components contain higher variance from the data

Can be used as FILTERING, by selecting only the top significant PCs

- PCs that explain at least 1% of variance
- The first 5-10 PCs

Thank you. Questions?

Paulo Czarnewski