Department of Biology
Penn
University of Pennsylvania

Introduction to Bioinformatics

ITMAT
ioinformatics

Topic 14
Dimensionality Reduction

Fall, 2023

Professor
Gregory R. Grant

Gregory R. Grant

Genetics Department

ggrant@pennmedicine.upenn.edu

ITMAT Bioinformatics Laboratory

University of Pennsylvania

Teaching Assistants
Chetan Vadali
Jianing Yang

Dimensionality Reduction

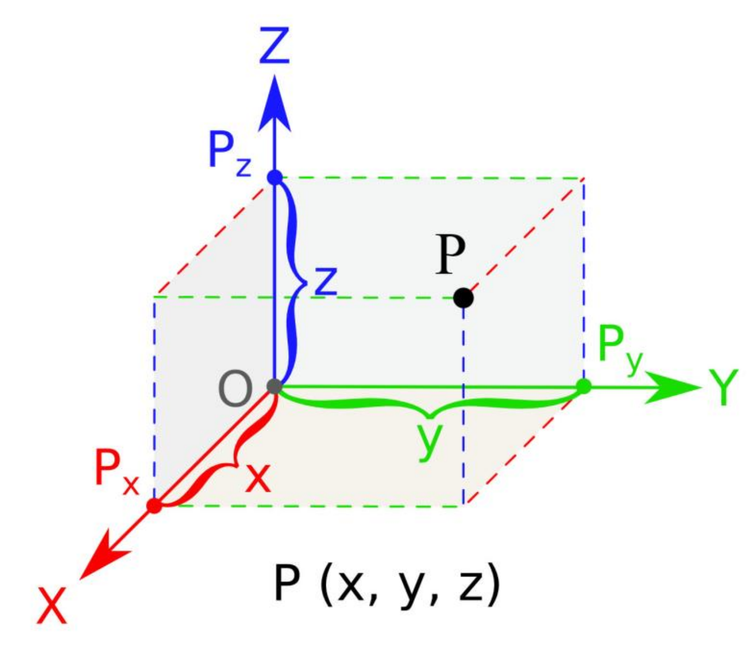
- We live in a (spatially) 3-dimensional world.
- We have little to no visual intuition for the 4th dimension or higher.
- Even though we have no trouble describing such spaces mathematically.
- If ℝ is the real numbers, then the 3rd dimension is simply the set:

$$\{(x, y, z) \mid x, y, z \epsilon \mathbb{R}\}$$

- In other words, the set of all triples of real numbers.
- And therefore the 4th dimension is:

$$\{(x, y, z, w) \mid x, y, z, w \epsilon \mathbb{R}\}$$

- The set of all 4-tuples of real numbers.

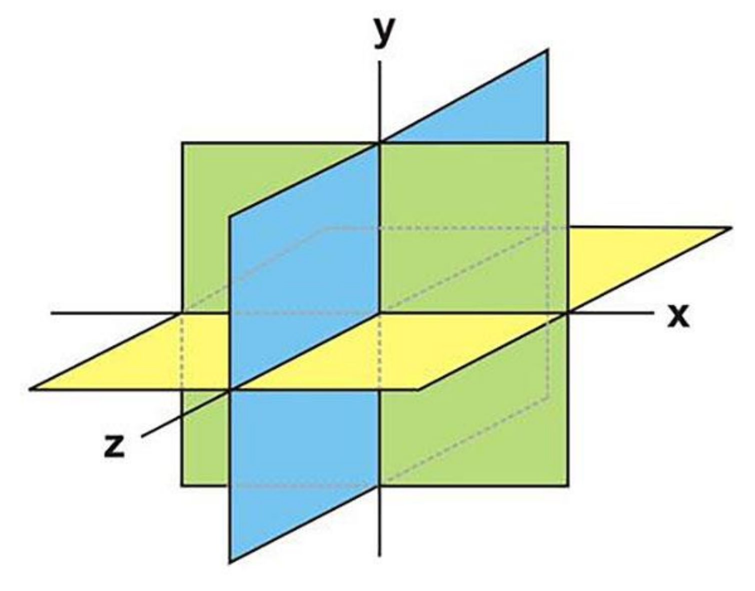


The Canonical Subspaces

- We call 3-dimensional space $\mathbb{R}^3$
  $$\mathbb{R}^3 = \{(x, y, z) \mid x, y, z \in \mathbb{R}\}$$

- A ***subspace*** of $\mathbb{R}^3$ is a line or a plane that includes the origin.

- There are three *natural* 2-dimensional subspaces in $\mathbb{R}^3$.
  – The X-Y plane
  – The X-Z plane
  – The Y-Z plane



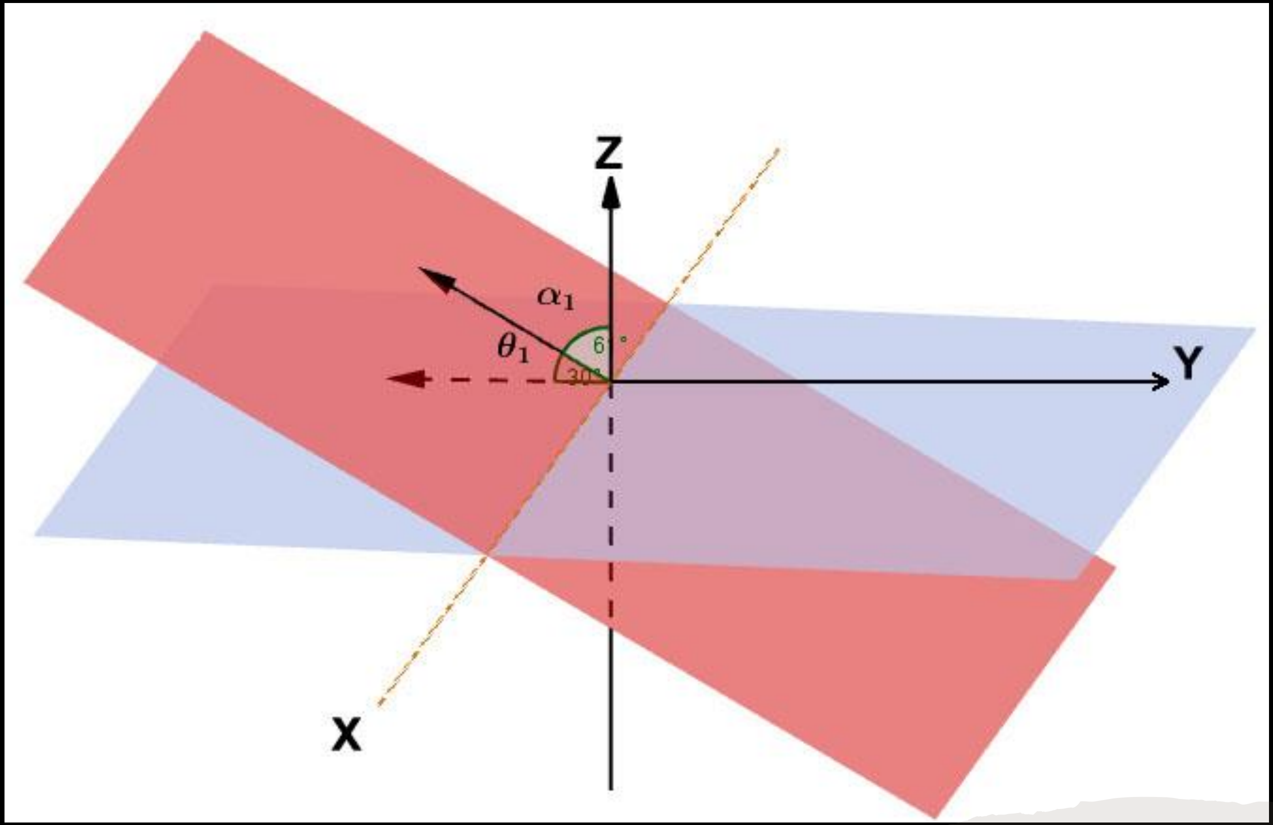
The X-Y plane is $\{(x, y, 0) \mid x, y \in \mathbb{R}\}$

The Y-Z plane is $\{(0, y, z) \mid y, z \in \mathbb{R}\}$

The X-Z plane is $\{(x, 0, z) \mid x, z \in \mathbb{R}\}$

Other Subspaces

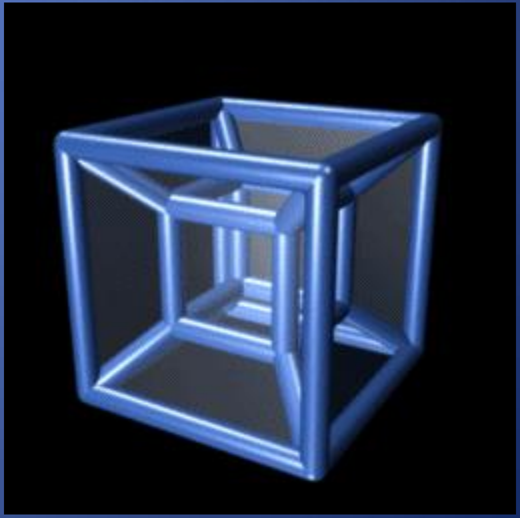
- And there are infinitely many other possible 2-dimensional subspaces.
- For example, $\{(x, y, z) \mid x, y, z \epsilon \mathbb{R}\ and\ x + y + z = 0\}$



Subspaces of the 4th Dimension

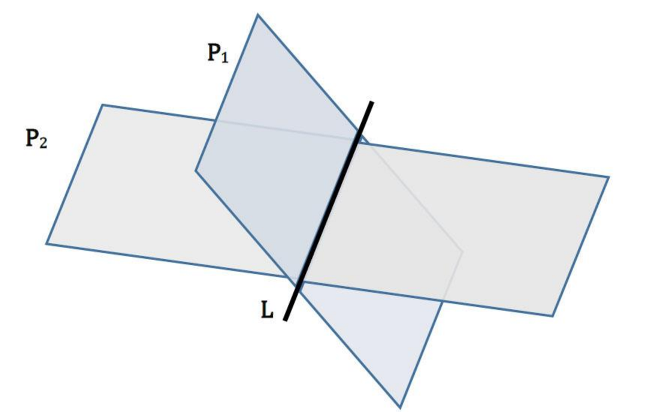
- We can't draw pictures, but we can still describe them mathematically.
- 4-dimensional space $\mathbb{R}^4$ is just the set of all quadruples of real numbers
$$\mathbb{R}^4 = \{(x, y, z, w) \mid x, y, z, w \epsilon \mathbb{R}\}$$

- There are six *natural* 2-dimensional spaces in $\mathbb{R}^4$.
  - The X-Y plane is $\{(x, y, 0,0) \mid x, y \epsilon \mathbb{R}\}$
  - The X-Z plane is $\{(x, 0, z, 0) \mid x, z \epsilon \mathbb{R}\}$
  - The X-W plane is $\{(x, 0,0, w) \mid x, w \epsilon \mathbb{R}\}$
  - The Y-Z plane is $\{(0, y, z, 0) \mid y, z \epsilon \mathbb{R}\}$
  - The Y-W plane is $\{(0, y, 0, w) \mid y, w \epsilon \mathbb{R}\}$
  - The Z-W plane is $\{(0,0, z, w) \mid z, w \epsilon \mathbb{R}\}$



Intuition in the 3rd Dimension

- We will soon be working in 30,000th dimensional space.
- But intuition already flies out the window in the 4th dimension.
- For example, in the 3rd dimension, how do two different planes intersect?
- Intuition tells us they must intersect in a straight line, or not at all.



Intuition in the 4th Dimension

- Same question.
  - In the 4th dimension, how do two planes intersect?

- It could still be a straight line.
  - The X-Y and Y-Z subspaces intersect in the Y-axis.

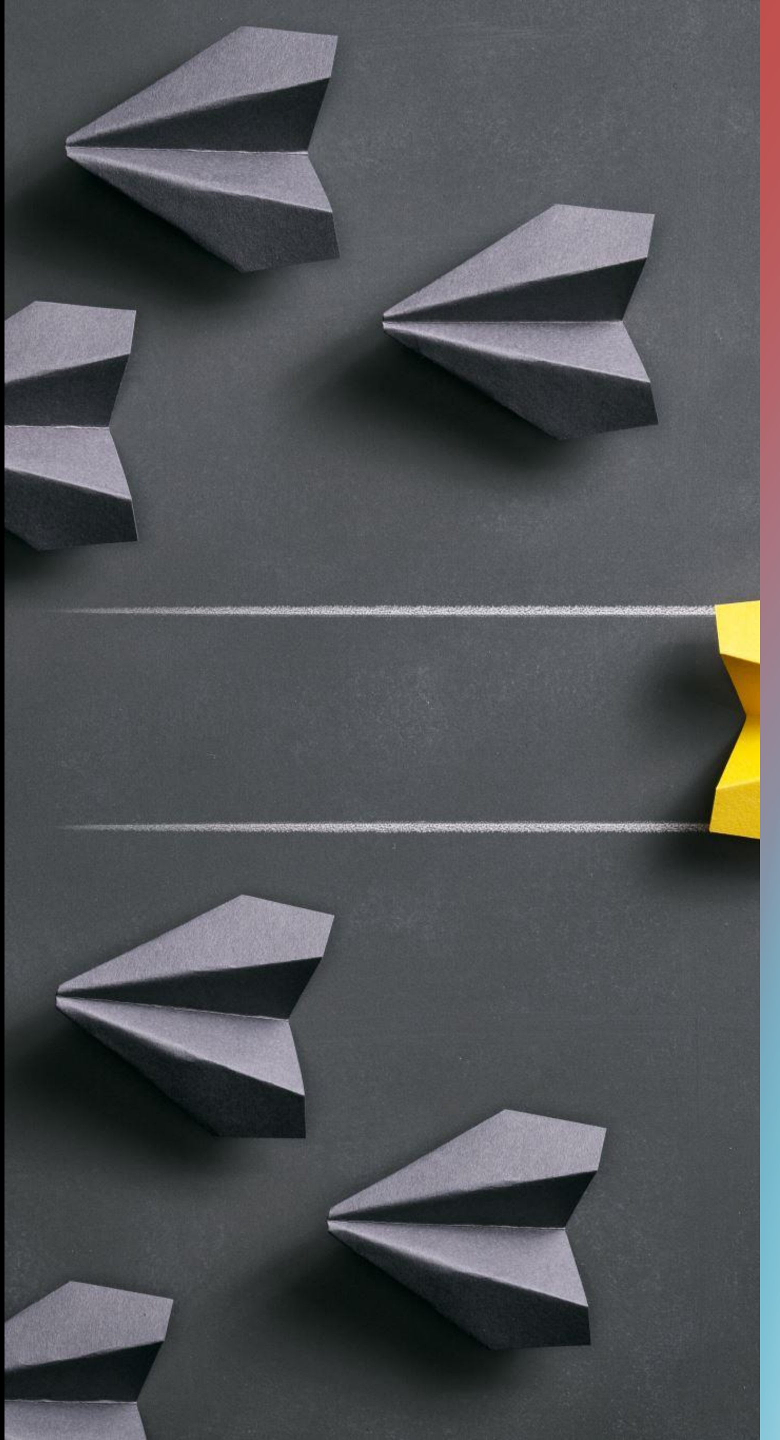
- But it’s also possible for two planes to intersect in one single point.

Two Planes That Intersect in One Point

- The X-Y and Z-W subspaces intersect only at the origin (0,0,0,0).
  - The X-Y plane is $\{(x, y, 0,0) \mid x, y \epsilon \mathbb{R}\}$
  - The Z-W plane is $\{(0,0, z, w) \mid z, w \epsilon \mathbb{R}\}$

- The only point in both sets is (0,0,0,0).

- This is obvious and uncontroversial when we write down the math like this.
- Yet it seems crazy and impossible at the level of intuition.



High-Dimensional Space

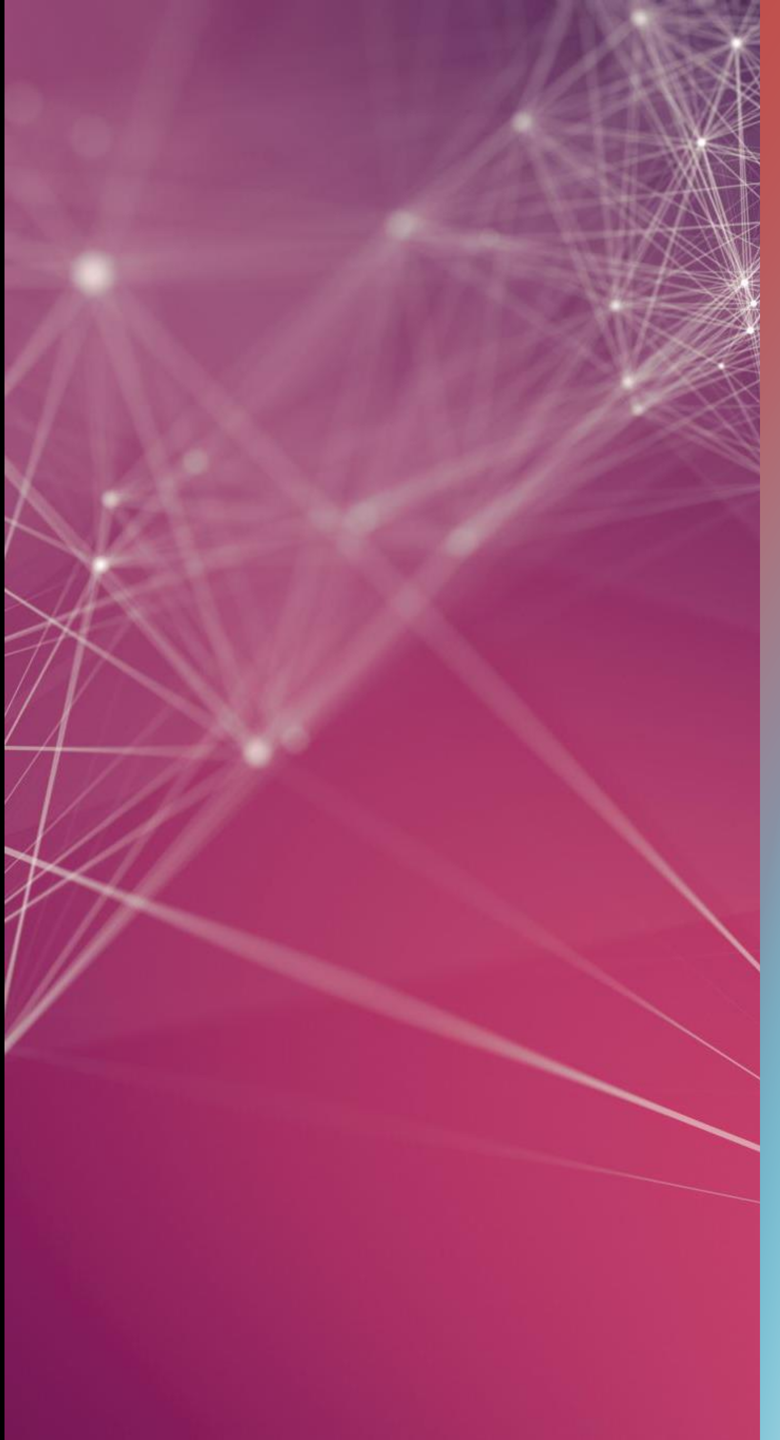
- and gene expression data -

- A triple of numbers (x, y, z) is a single point in 3-dimensional space.
  - We also call a triple of numbers a vector of length 3.

- And a quadruple of numbers (x, y, z, w) is a single point in 4-dimensional space.
  - A vector of length 4.

- Well, what is the collection of gene quantifications associated to an RNA-Seq sample?
  - It's a vector of length (approximately) 30,000

- Therefore, one RNA-Seq sample is *a single point* in 30,000-dimensional space.



Notation

- We quickly run out of letters so in higher-dimensional space we use subscripts

$$\left(x_1, x_2, \dots, x_{30{,}000}\right)$$

- Any letter can be used.

$$\left(y_1, y_2, \dots, y_{30{,}000}\right)$$

- Or if we have 30,000 genes and *m* samples, we might use double subscripts, one for sample and another for gene.

Sample 1: $\left(x_{1,1}, x_{1,2}, \dots, x_{1,30{,}000}\right)$

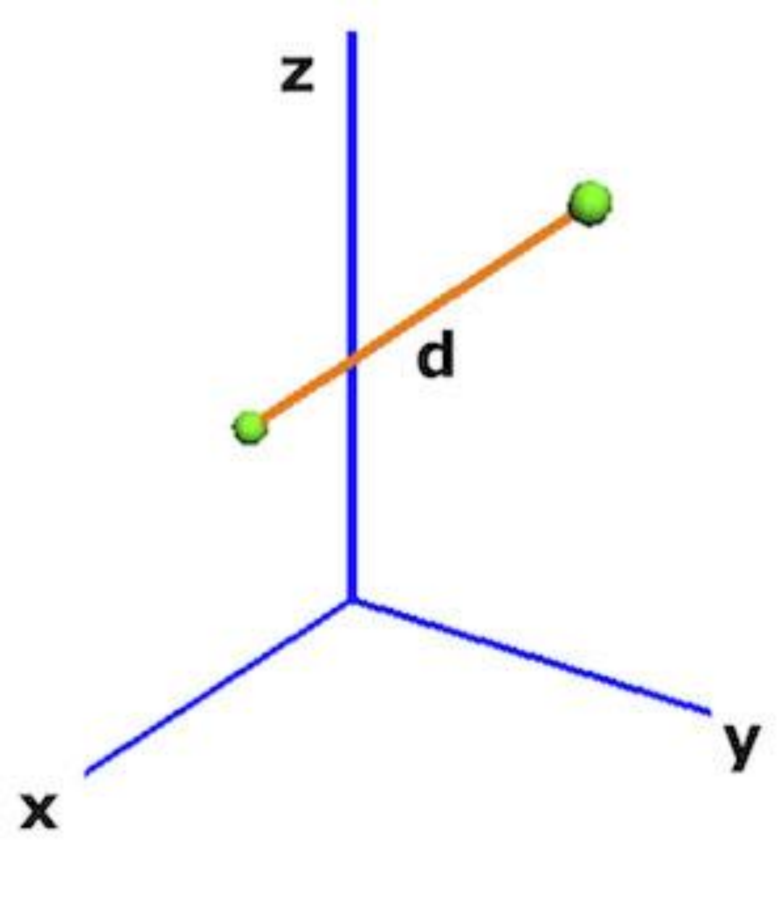
Sample 2: $\left(x_{2,1}, x_{2,2}, \dots, x_{2,30{,}000}\right)$

Sample 3: $\left(x_{3,1}, x_{3,2}, \dots, x_{3,30{,}000}\right)$

- In general, the *j*-th gene of the *i*-th sample is $x_{i,j}$

Distance

- If two samples have similar gene expression, then their corresponding points in 30,000-dimensional space should be physically near each other.
- These distances are of interest because they could reveal hidden relationships between the samples.
- This is akin to hierarchical clustering but represents the information differently.
  - In space rather than by a tree.



Distances Formula

- We can calculate distances in 30,000-dimensional space using the usual formula.
- If $\vec{x} = (x_1, \dots, x_{30,000})$ and $\vec{y} = (y_1, \dots, y_{30,000})$
- Then the (Euclidean) distance between $\vec{x}$ and $\vec{y}$ is

$$d(\vec{x}, \vec{y}) = \sqrt{\sum (x_i - y_i)^2}$$

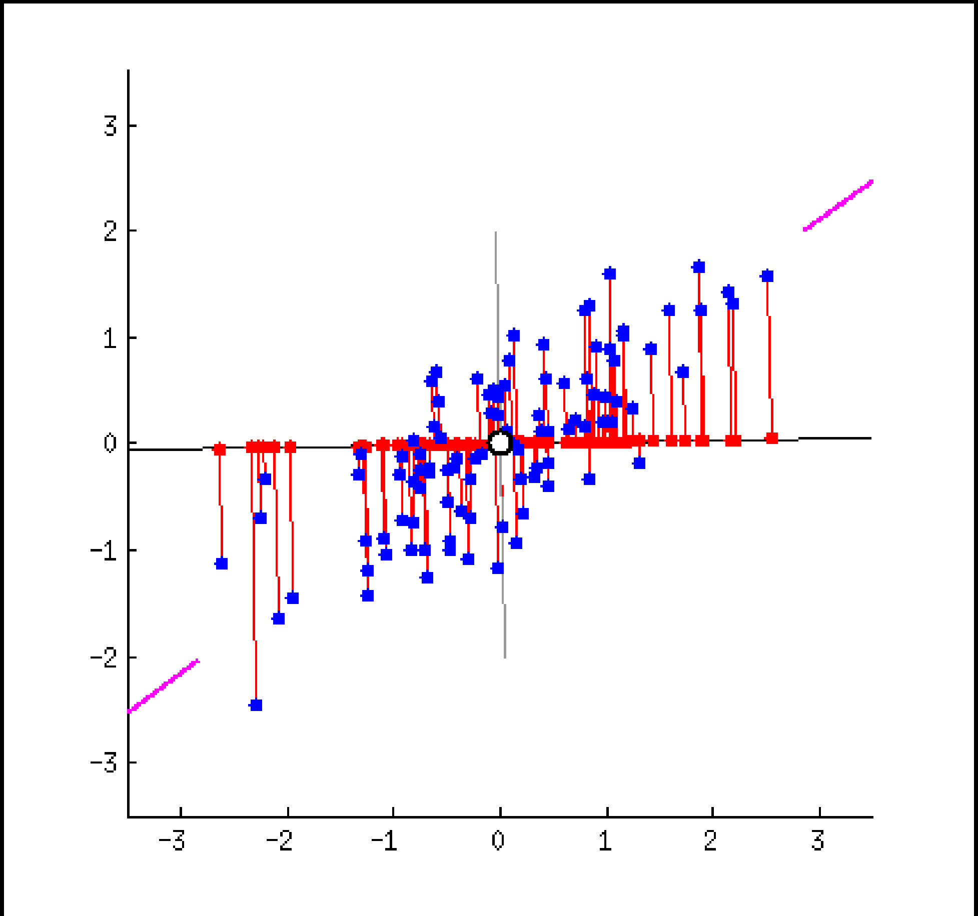
Numbers versus Pictures

- We can easily calculate the distances between all points (samples) and put them in a matrix.
- But staring at a matrix of numbers is only so enlightening.
- We’d prefer to visualize the points like we would if they were in 2-dimensional or 3-dimensional space.

Projections

- We achieve this by “projecting” the data from the 30,000-dimension to the 2nd dimension.

- However, to gain some intuition of what a projection is, let’s start by projecting points from 2-dimensional space onto one-dimensional subspaces.

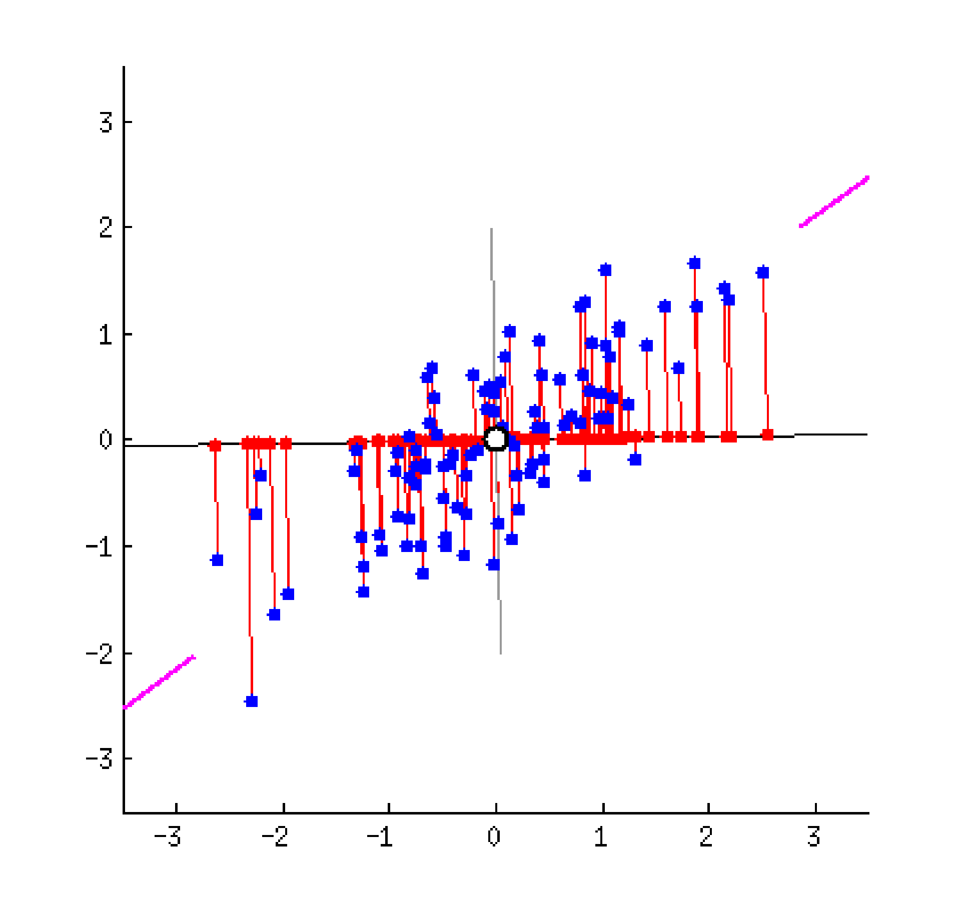


Not All Projections Are Equal

- Notice that some projections crunch all the data together after projections
  - Subspaces with slope around -1

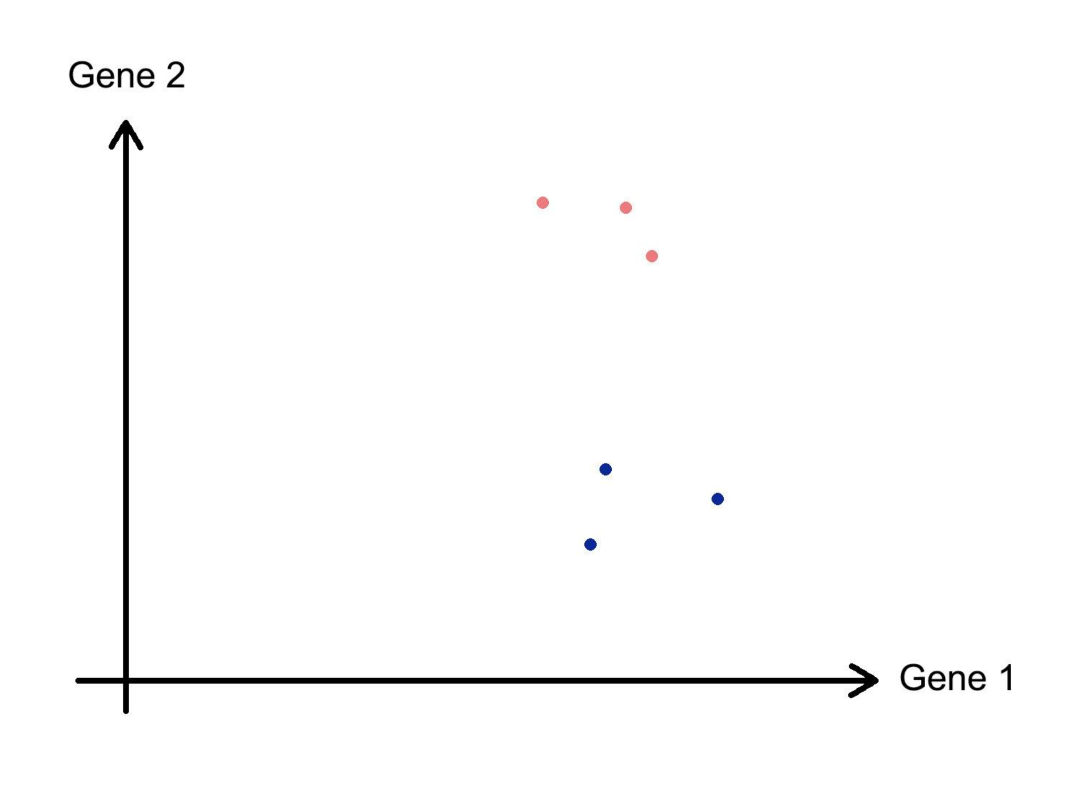
- And others keep them relatively spread out.
  - Subspaces with slope around +1

- The more spread out the data are in the subspace, the more information about their relationships is preserved.



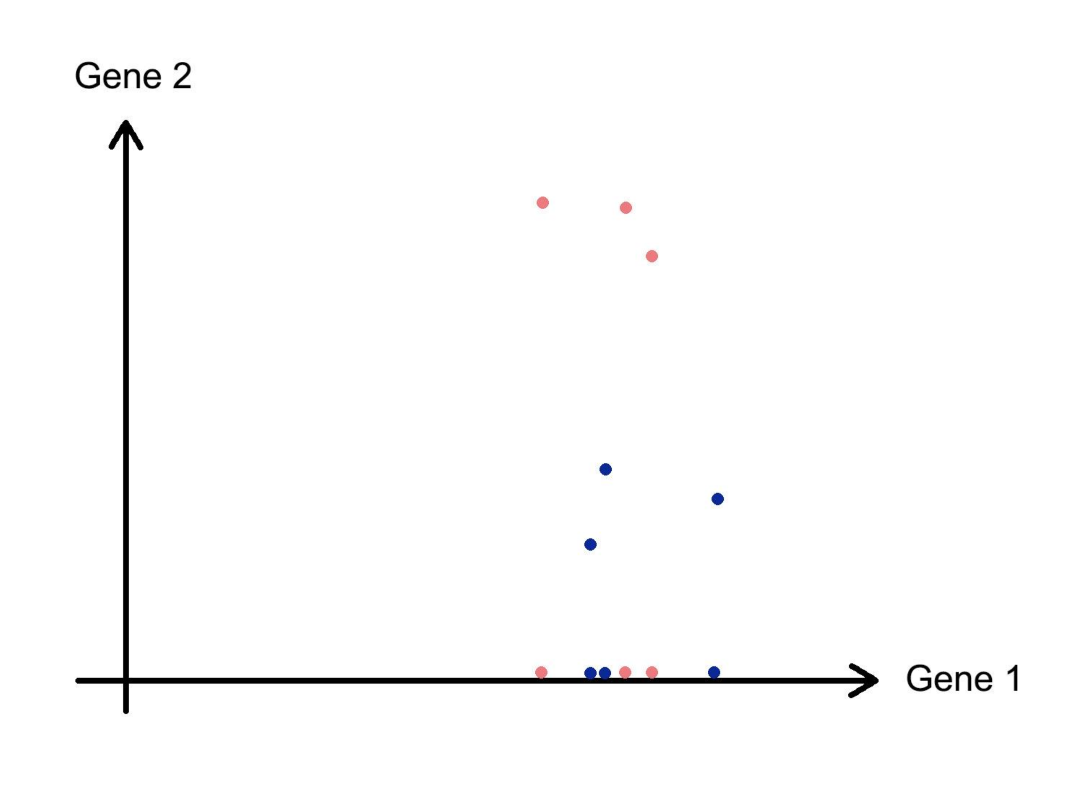
Baby Example

- Imagine there were only two genes.
- This is gene expression data for six subjects.
- Visually, Gene 2 is DE, Gene 1 is not.



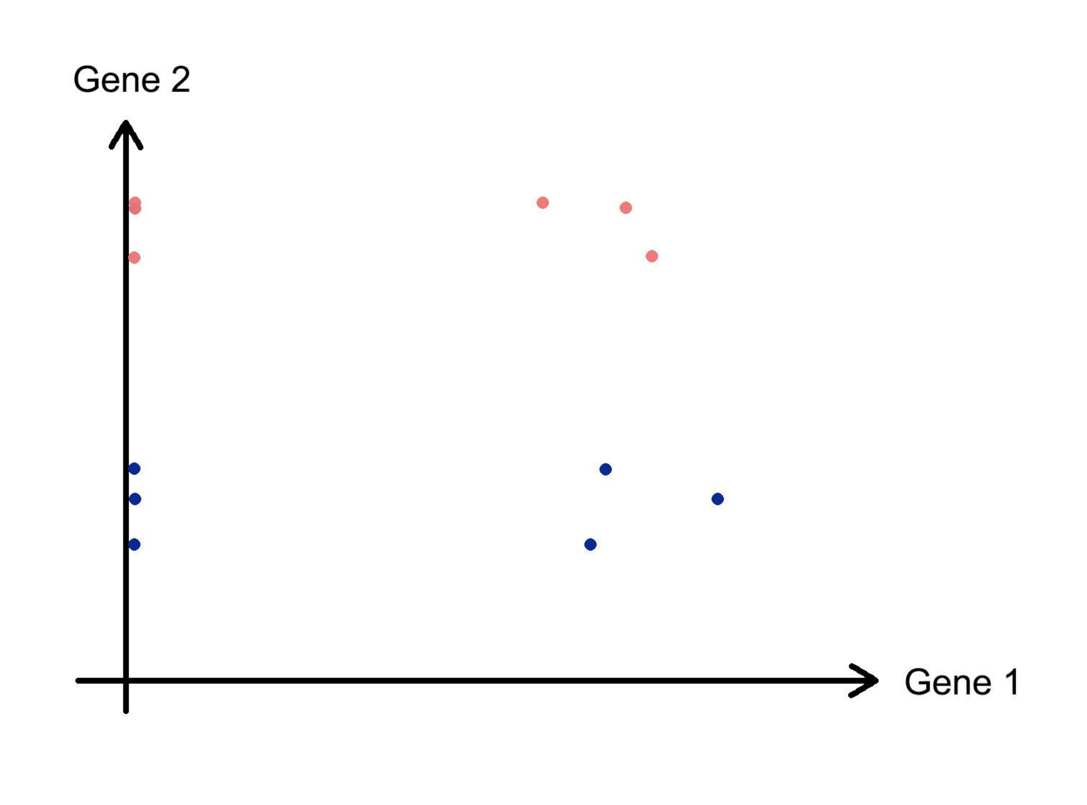
Projection onto X-Axis

- This is the same as just forgetting Gene 2.
- Imagine you can't see in two dimensions; all you can see is the X-axis.
- Since Gene 1 is not DE, we lose the separation.



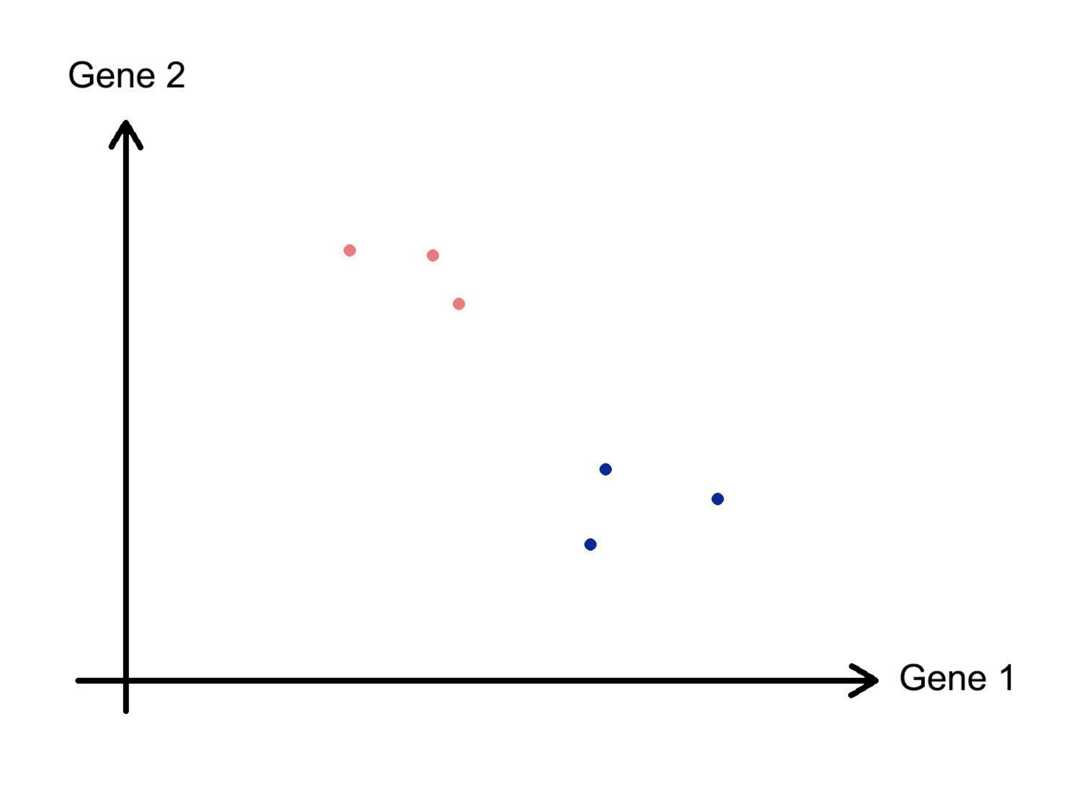
Projection onto Y-axis

- This is like forgetting Gene 1.
- Imagine you can't see in two dimensions; all you can see is the Y-axis.
- Since Gene 2 is DE, this preserves the separation.



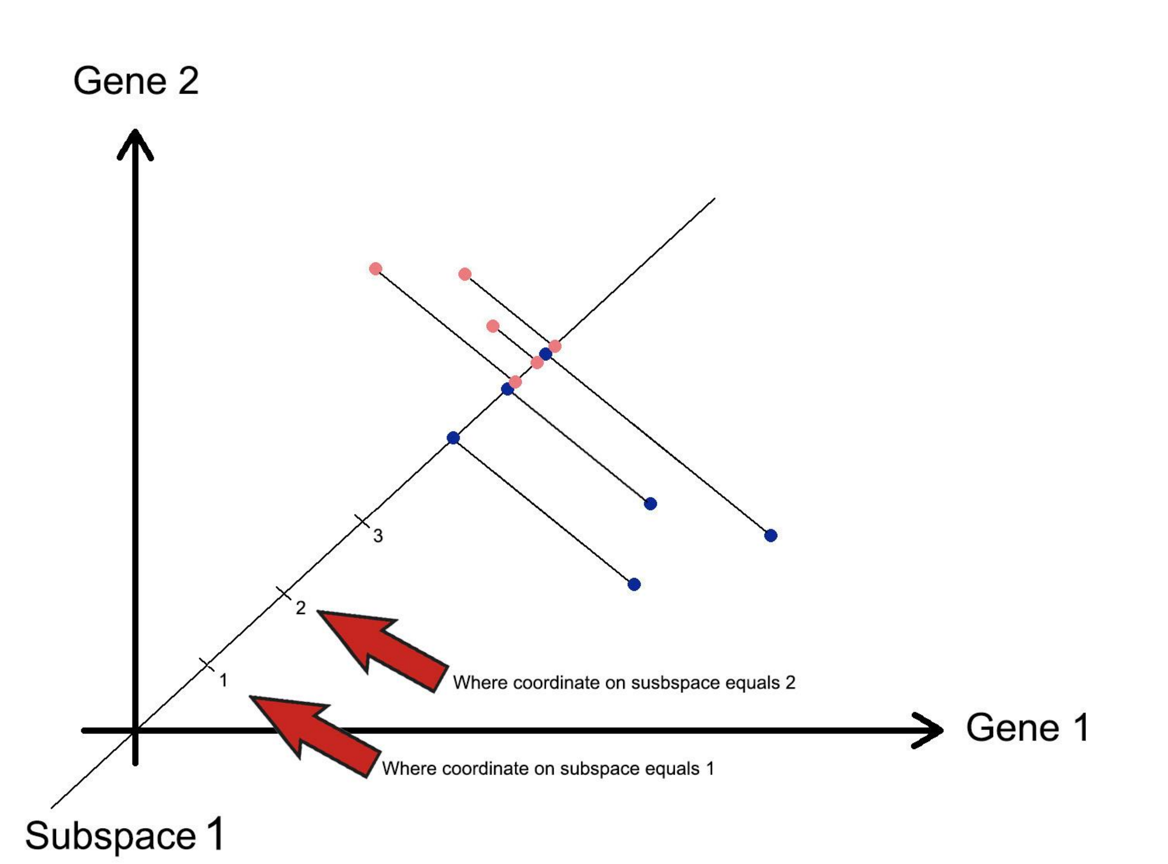
Other Subspaces

- Now suppose the data looks like this.
- Now it appears both genes are DE.
- But we're going to continue to imagine we can't see in two dimensions, only one.



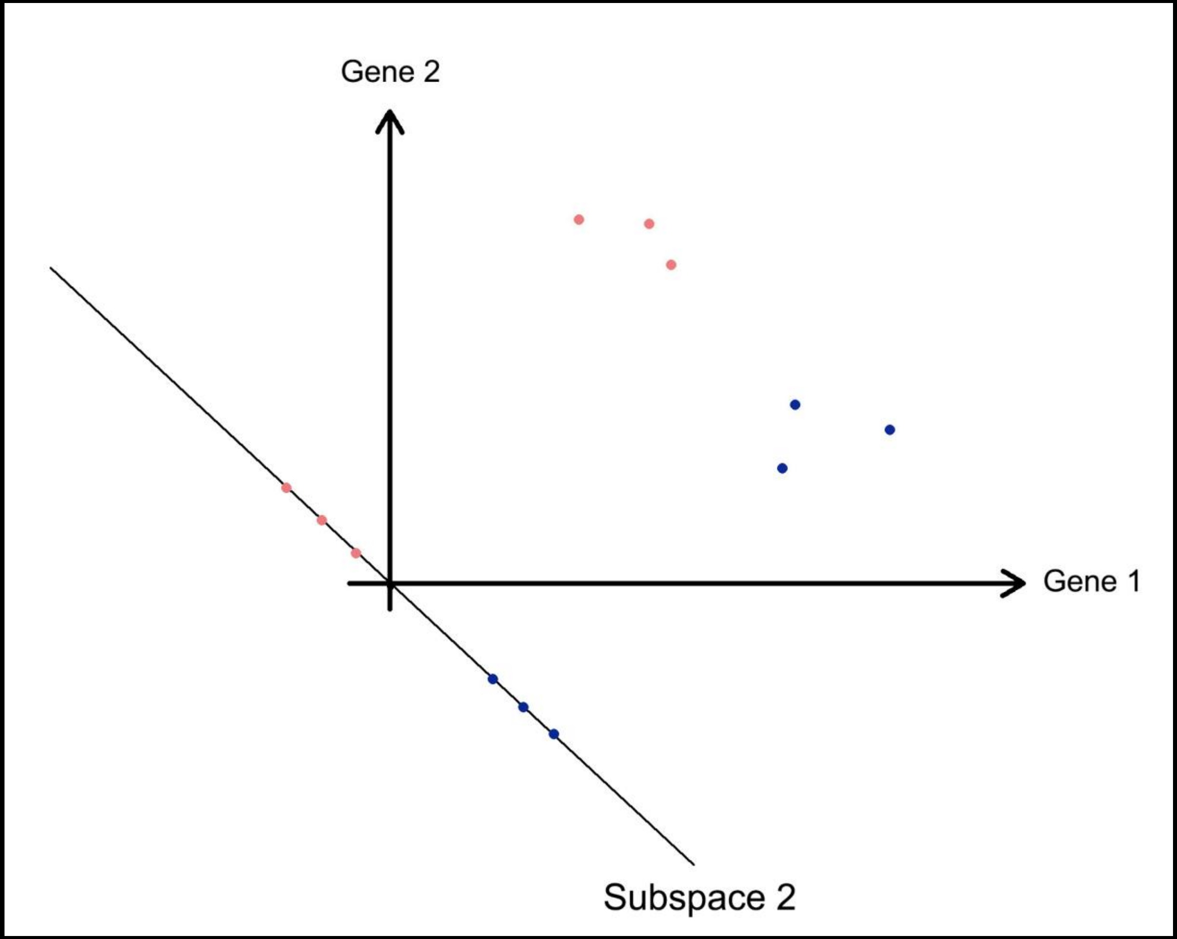
Diagonal Subspace

- This subspace can be thought of as an equal combination of both genes.
- Strictly speaking, every subspace is a *weighted* sum of the two axes.
- Here both weights are equal to one, thus the perfect 45° diagonal.


Gene 2
Gene 1
Subspace 1
1
2
3
Where coordinate on susbspace equals 2
Where coordinate on subspace equals 1

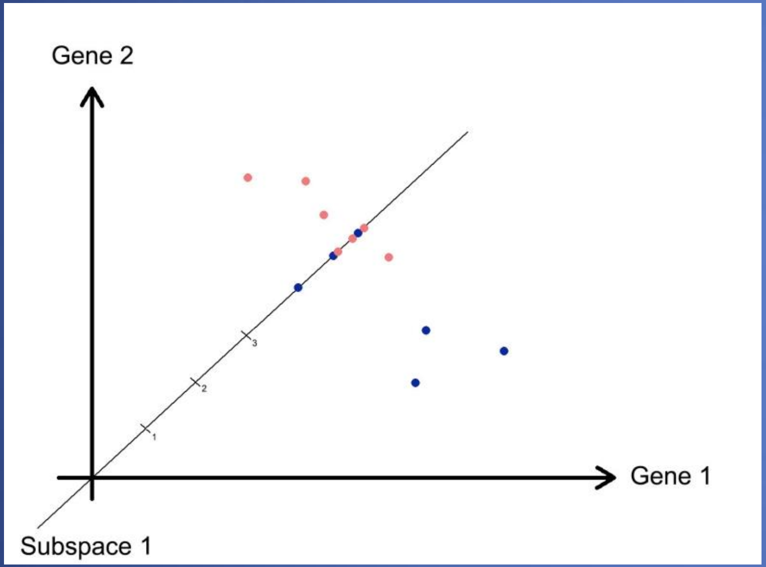
The other Diagonal Subspace

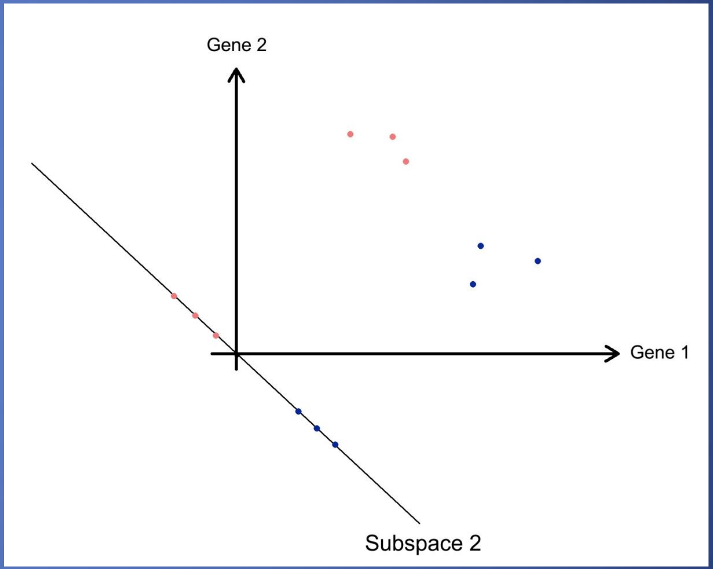
- The subspace at -45° is the weighted sum where the weights are +1 and -1.

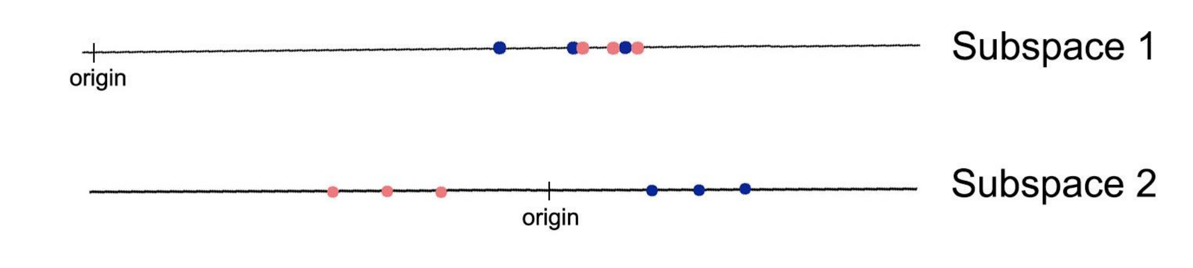


The 1D view

- We can draw the subspaces as 1D spaces on their own.
- Here we’ve drawn two, of the infinitely many possible.

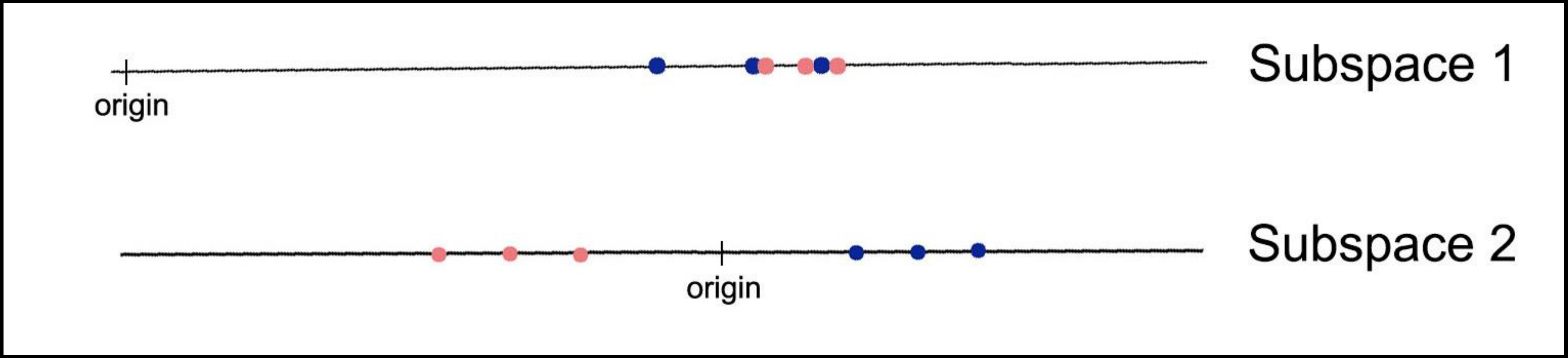






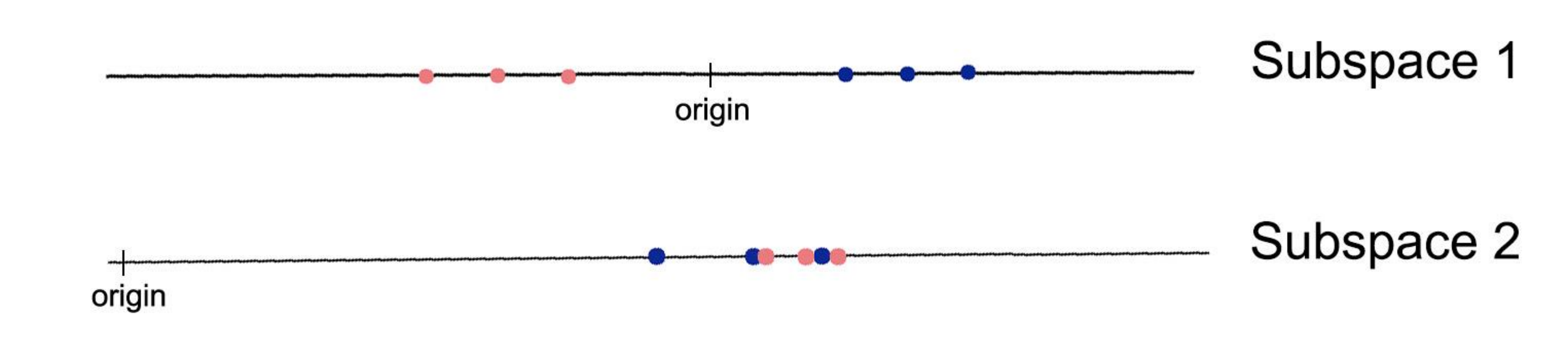
The view from one dimension

- Continuing to assume we’re one-dimensional creatures who can only visualize in one dimension, these “slices” are our *only* way to view of the 2D data.
- And if all we looked at was subspace 1, then we would not see the separation.
- The data has much greater variance in Subspace 2.
- This is why the dimensionality reduction algorithms look for the subspace with the greatest variance.



Latent Variables

- Instead of working with two variables, Gene 1 and Gene 2, we can just work with the values projected onto Subspace 1
  - This has retained much of the information from the two genes, but we only have to work with one variable instead of one.
- Subspaces are called “latent variables”.
  - And there are infinitely many of them, one for each possible weighted sum.
- By choosing latent variables judiciously, we can work with fewer variables.
- And ultimately, to work with RNA-Seq data, we need to get that down from 30,000 to two.



Interpreting Latent Variables

- Each latent variable is a weighted sum of the expression values of all genes.

$$L = \sum_i w_i G_i$$

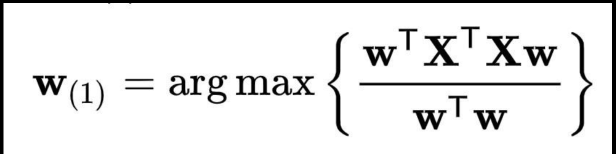
  - L is the value of the latent variable.
  - G_i is the expression value of gene i
  - w_i is the weight for gene i

- If all weights are zero except for the genes in some particular pathway, then the latent variable is all about that pathway. No mystery.
- And sometimes they're more mysterious, when they are not focused on one pathway or functional category.

Variance

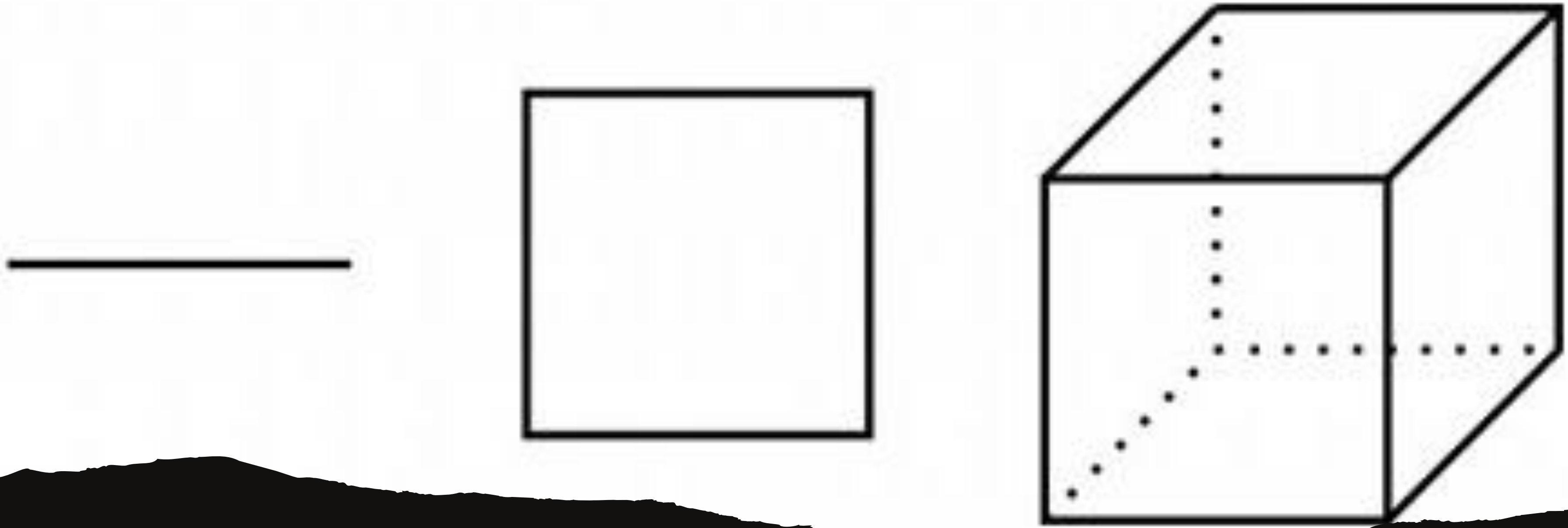
- As the previous slides make clear, we want to project on the subspace that preserves the maximum amount of variance in the data.

- There are elegant formulas from linear algebra (matrix algebra) that give this with surprisingly little work.



$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathbf{w}}{\mathbf{w}^{\mathsf{T}}\mathbf{w}} \right\}$$

Just showing what they look like here, you are not responsible for this formula.



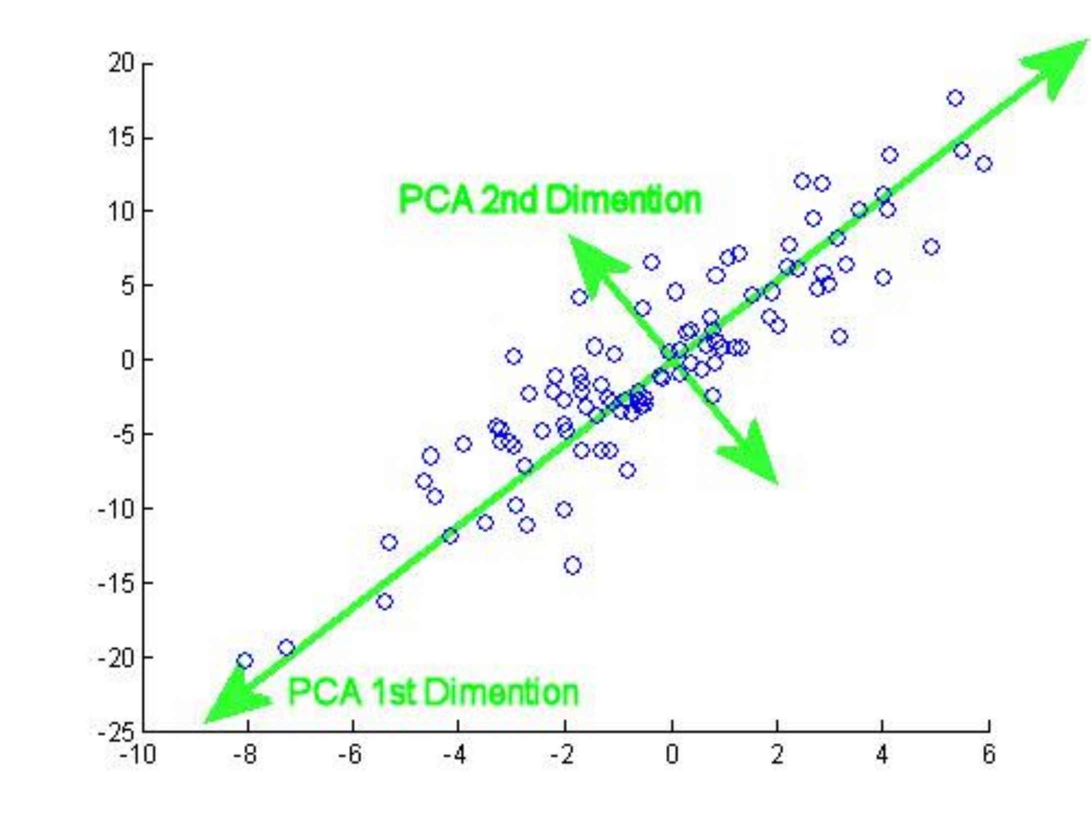
Two Dimensions

- We’ve been illustrating concepts by projecting onto one dimensional subspaces.
- But we can visualize two and also three dimensions.
- We typically project onto two dimensional subspaces because they are the easiest to render.

PCA1 and PCA2

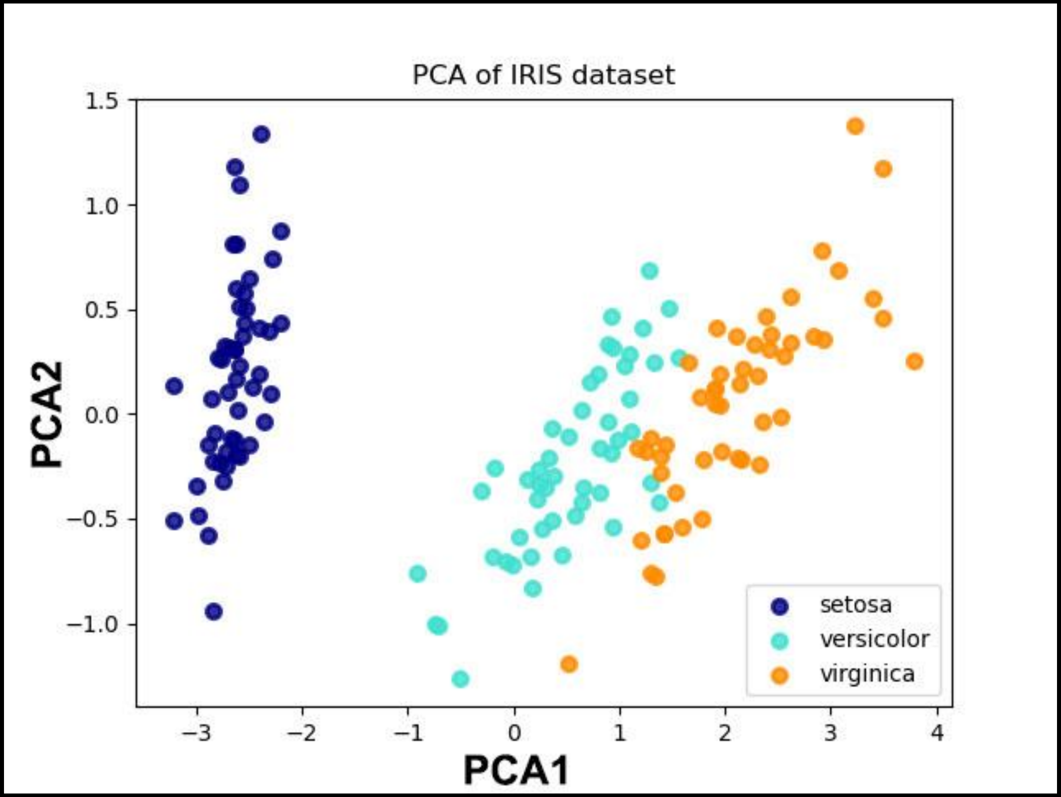
- PC1 is the 1-dimensional subspace which retains the greatest amount of variance when the data are projected onto it.

- PC2 is the 1-dimensional subspace *that is perpendicular to* PC1 and has the greatest amount of variance when the data are projected onto it.



Principal Components Analysis

- In PCA you start by projecting the original high-dimensional data onto the 2-dimensional subspace defined by the two one-dimensional subspaces PC1 and PC2.
- You then try to interpret what you see.



PCA3

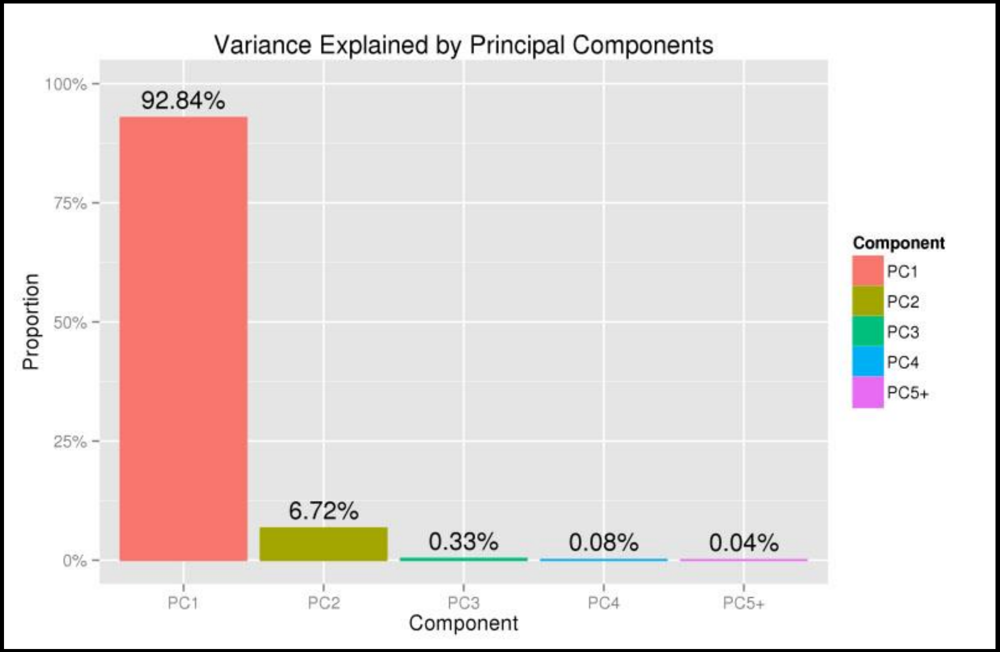
- Continuing, we can define PC3 as the (one-dimensional) subspace that’s perpendicular to the plane defined by PC1 and PC2 and retains the greatest amount of variation of the data among all such perpendicular subspaces.

- We can then graph PC1 or PC2 against PC3.
  - And nothing stops us from continuing to PC4, PC5, etc.
  - As long as we just plot one of these against one other, we get a 2-dimensional plot that we can visualize.

- But there are diminished returns going to higher and higher components.

Percent Variance Explained

- The data projected onto PC1 will (almost certainly) be less variable than the original data is the full 30K dimensional space.
  - PC1 is the subspace that preserves the most variance, but it can't usually preserve it all.
- Happily, Principal Components Analysis also reports how much variance is explained by each Principal Component.



In this example, it would be pointless to look beyond PC3 and probably even beyond PC2 and arguably just PC1 might be enough with 92% of variance explained just by that one PC.

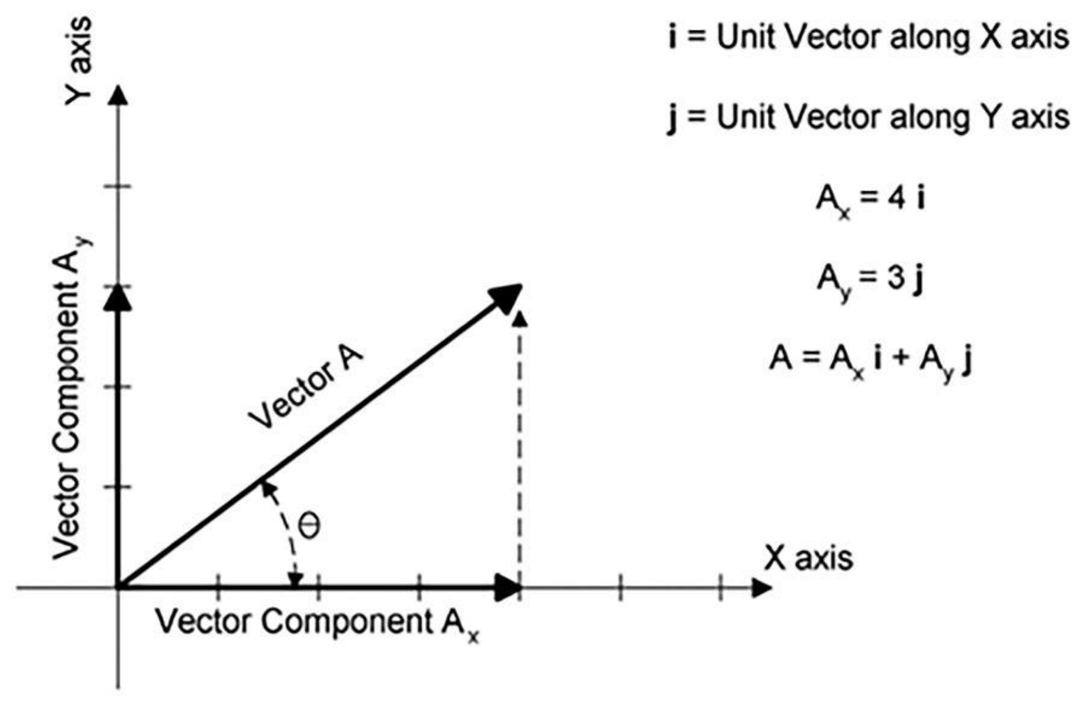
Loadings

- Each Principal Component is a weighted sum of the original dimensions.

- The weights w_i tell us how much that dimension (gene) contributes to the Principal Component.

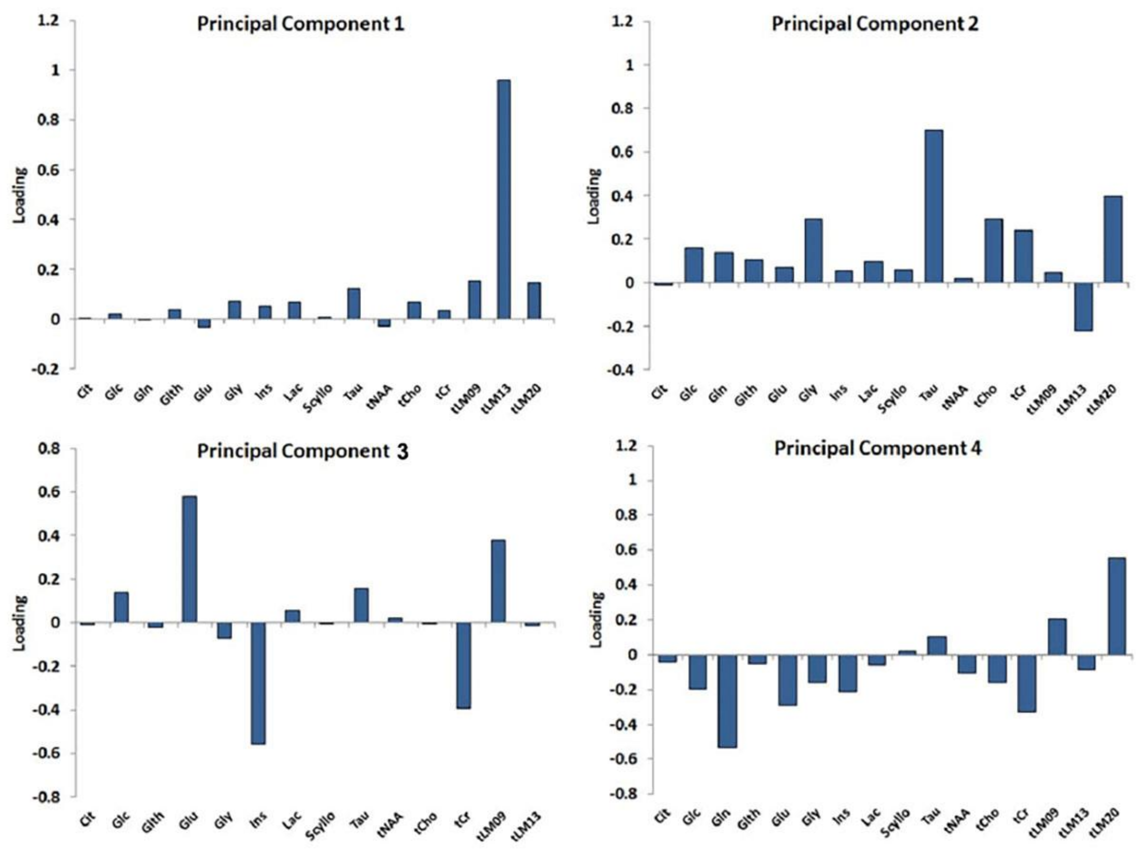
- For example, if all w_i were 0 except w_{85} then we'd know all variation in that component is explained by variation of the 85th gene.

- Or if all w_i were 0 except the genes in one pathway, then we'd know all variation between samples is explained by variation in that pathway.



$$PC1 = \sum w_i g_i$$

Loadings



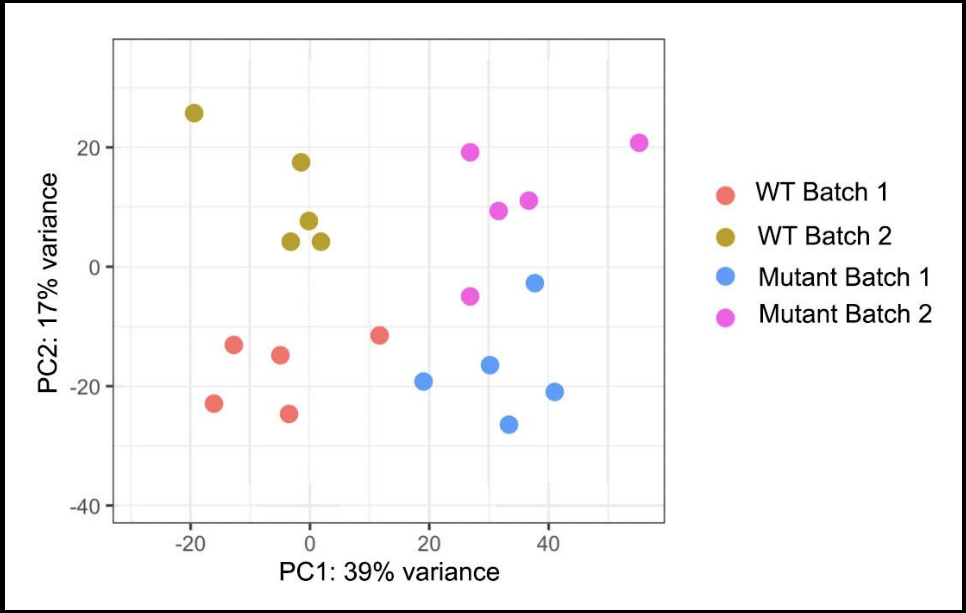
- Loadings can be represented in several ways.

- The most basic is as bar graphs where the weights have been normalized to be between -1 and +1.
  - Genes with loadings that are basically zero are omitted.

PCA Interpretation Summary

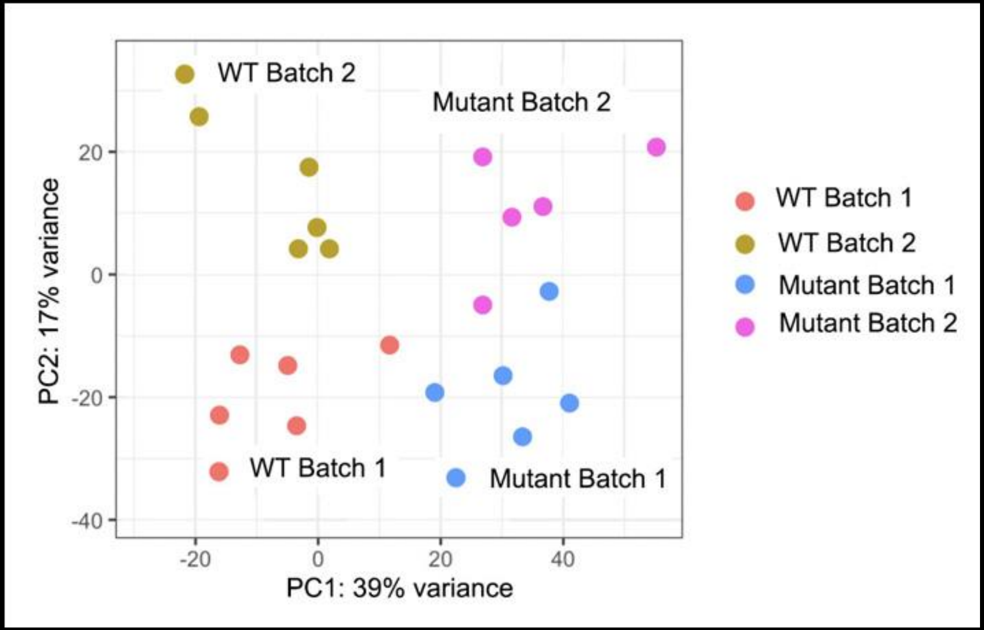
- To interpret PCA consider:
  - The percent variance explained by the various Principal Components.
    - This tells us how relevant the PC's are.
  - The loadings.
    - This tells us which genes are relevant to which PC's.
- Visually, look for clustering and groupings of points (samples).
  - We often color points by experimental conditions and other known factors to help with interpretation.

PCA Example



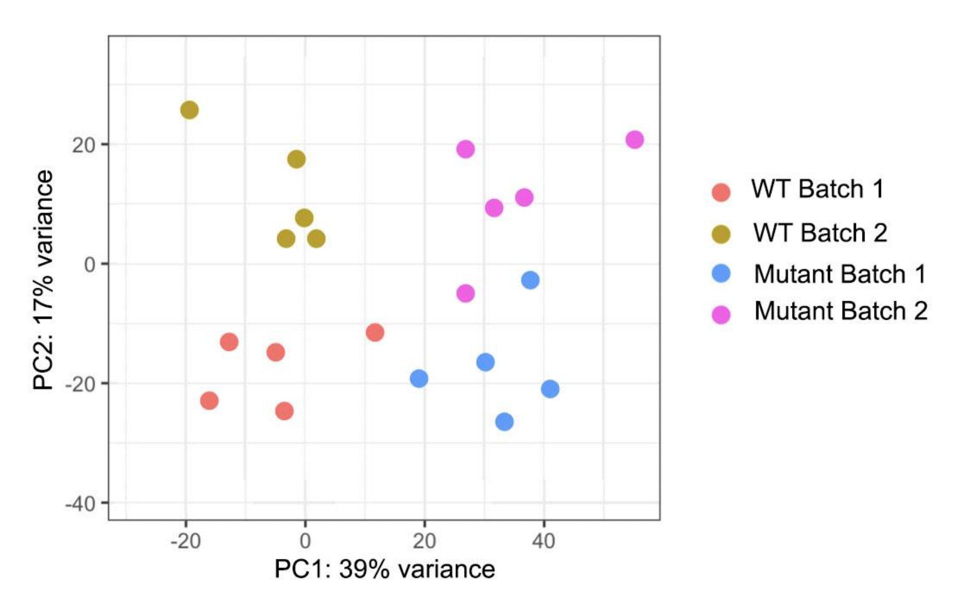
- Here WT and Mutant mice were compared.
  - And they were assayed in two batches.
- Think of “batch” and “genotype” as two (categorical) variables, each of which affect the variance of the assays.

PCA Example



- PCA shows batch had a lot of impact on expression.
  - PC2 is largely driven by batch while PC1 is driven by genotype.
- This indicates we should to account for batch in the analysis.
  - We can lower the technical variance.
- Since PC1 is driven by genotype, the loadings for PC1 indicate which genes are DE.
  - Focusing on PC1 effectively controls for batch in this data.

Controlling for Batch with Linear Model

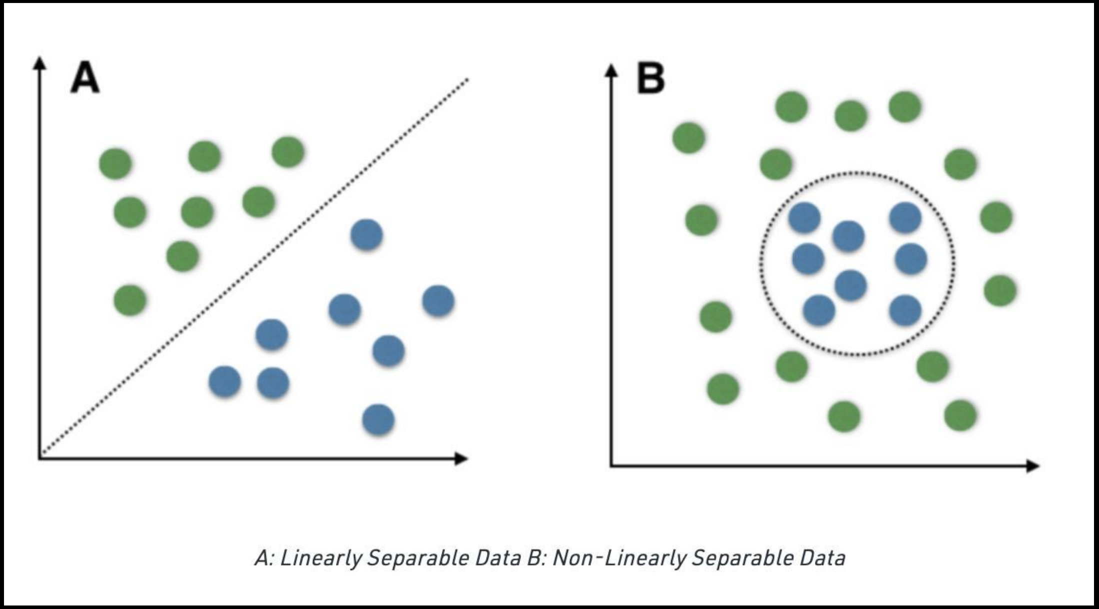


- Another way to control for batch is with a linear (regression) model that includes “batch” as a nuisance variable.

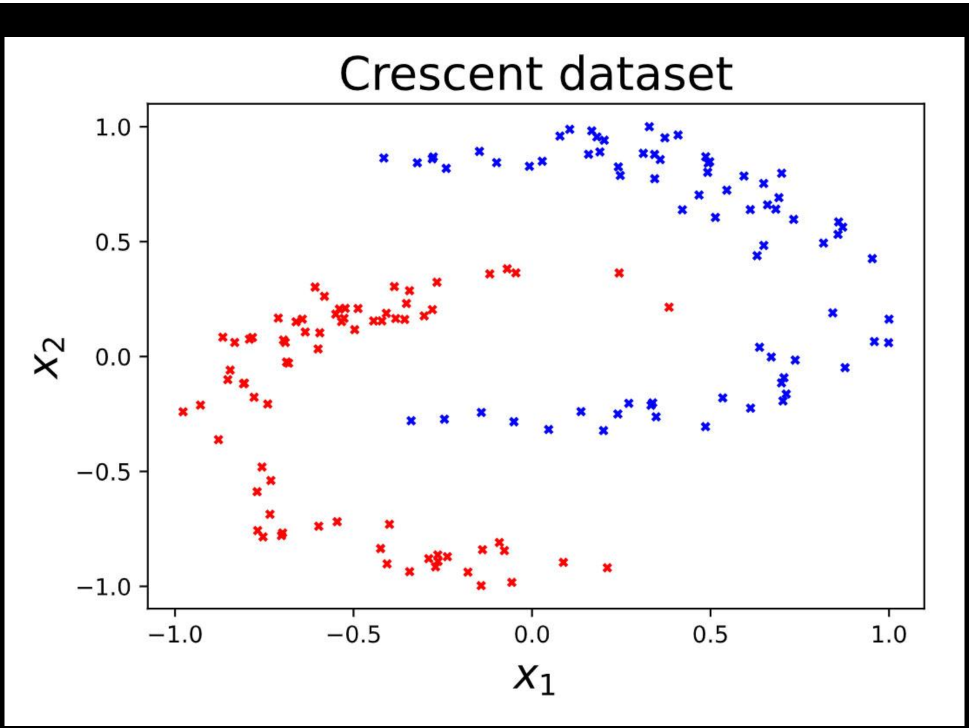
- Some off-the-shelf RNA-Seq DE apps are based on linear models and allow you to control for nuisance variables.
  - DESeq2 and Limma-voom for example.

Nonlinear dimensionality reduction

- PCA is known as a linear method.
  - The projection onto the subspace is performed by a linear transformation (in other words, multiplication by a matrix).
- But biological data are not always linearly separable.
- Any subspace we project onto linearly is going to mix the two conditions up.

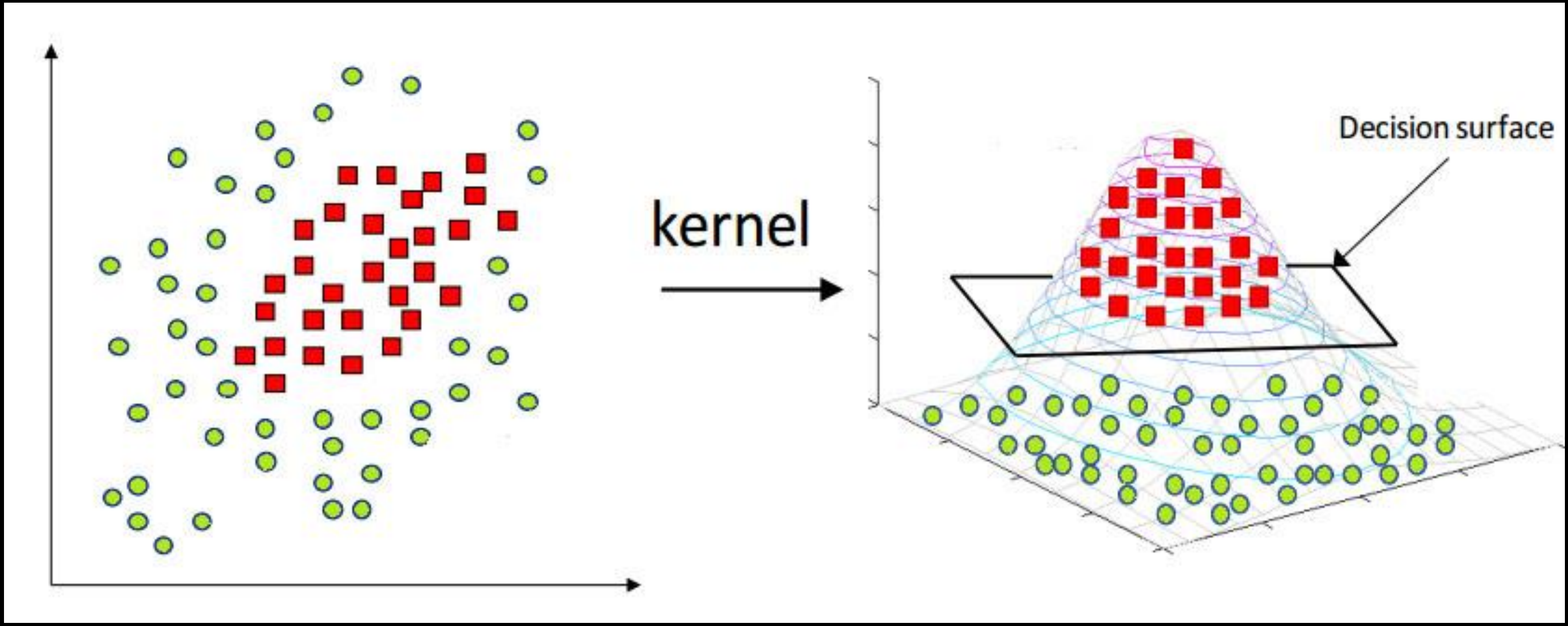


A: Linearly Separable Data B: Non-Linearly Separable Data



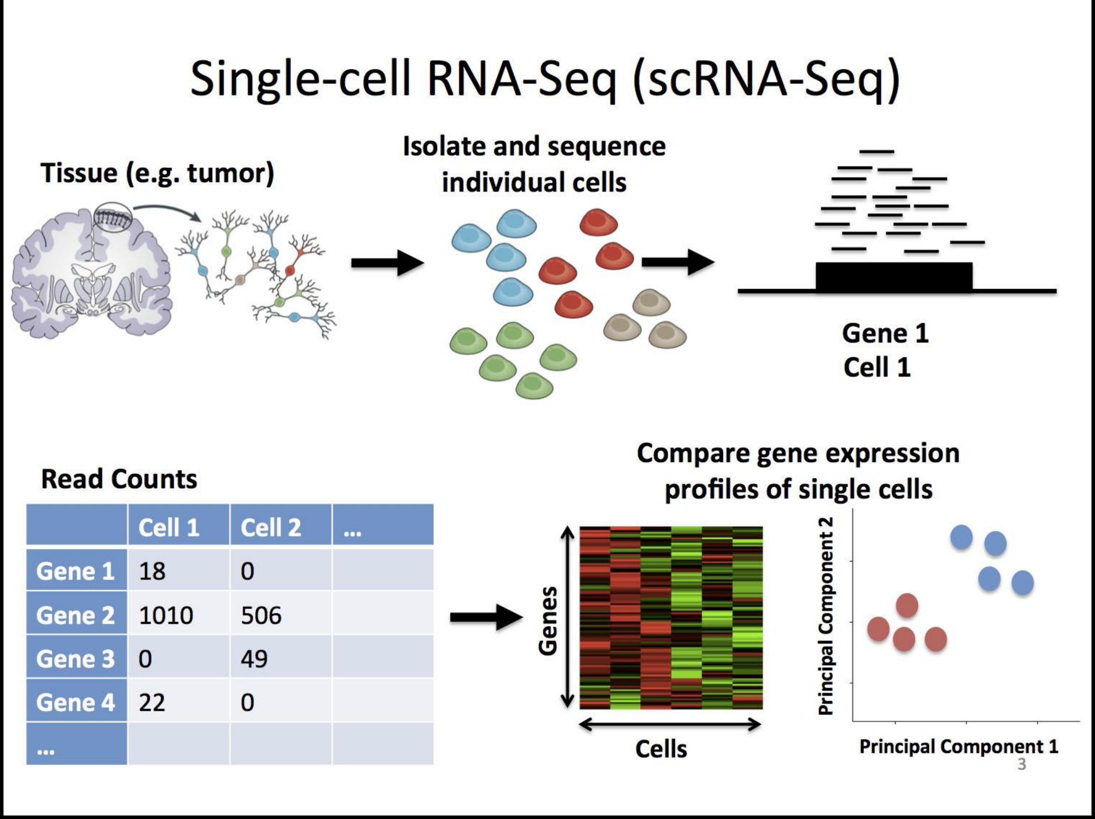
The Kernel Trick

- One way around this problem is to first embed into an even higher dimensional space, where the data then become linearly separable.



Single Cell Transcriptomics

- One of the most important domains for dimensionality reduction is single-cell transcriptomics.
  - Hundreds to thousands of cells are assayed by RNA-Seq.
  - Depth of sequencing is 100 times less than with bulk RNA-Seq.
  - But you get many cells.



Single-cell RNA-Seq (scRNA-Seq)

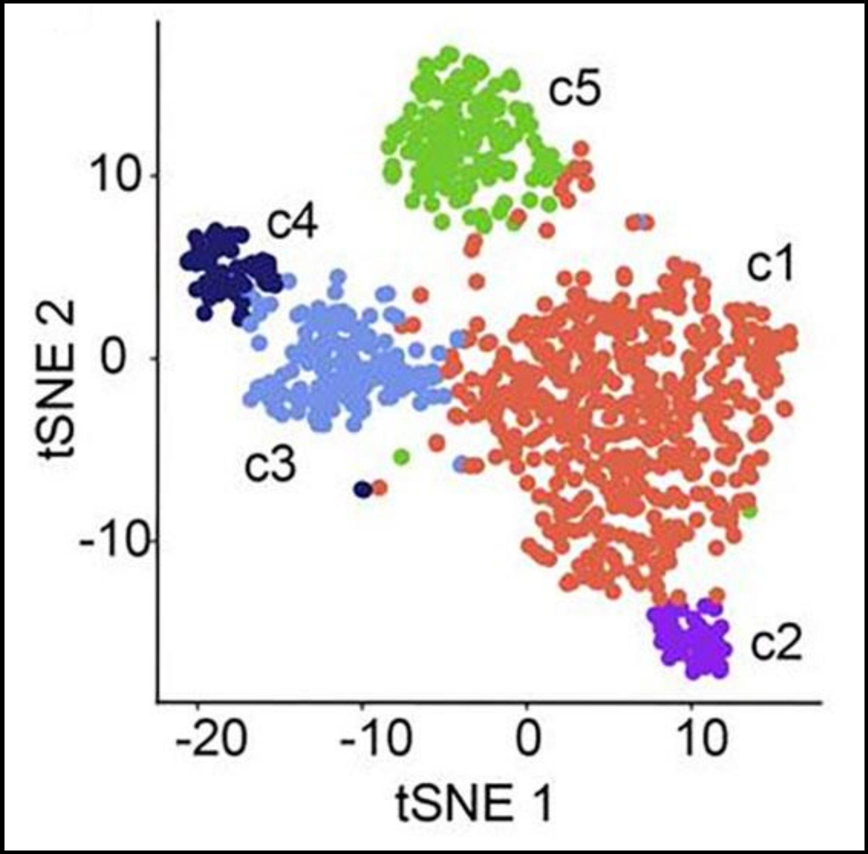
Tissue (e.g. tumor)
Isolate and sequence individual cells
Gene 1
Cell 1
Compare gene expression profiles of single cells
Genes
Cells
Principal Component 2
Principal Component 1
3

Read Counts

	Cell 1	Cell 2	...
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
...			

Single Cell Transcriptomics

- Dimensionality reduction allows us to identify types of cells based entirely on their transcriptome.
  - Even though they may be indistinguishable morphologically or by cell surface markers.
- It is generally accepted that non-linear dimensionality reduction performs better on single cell data.

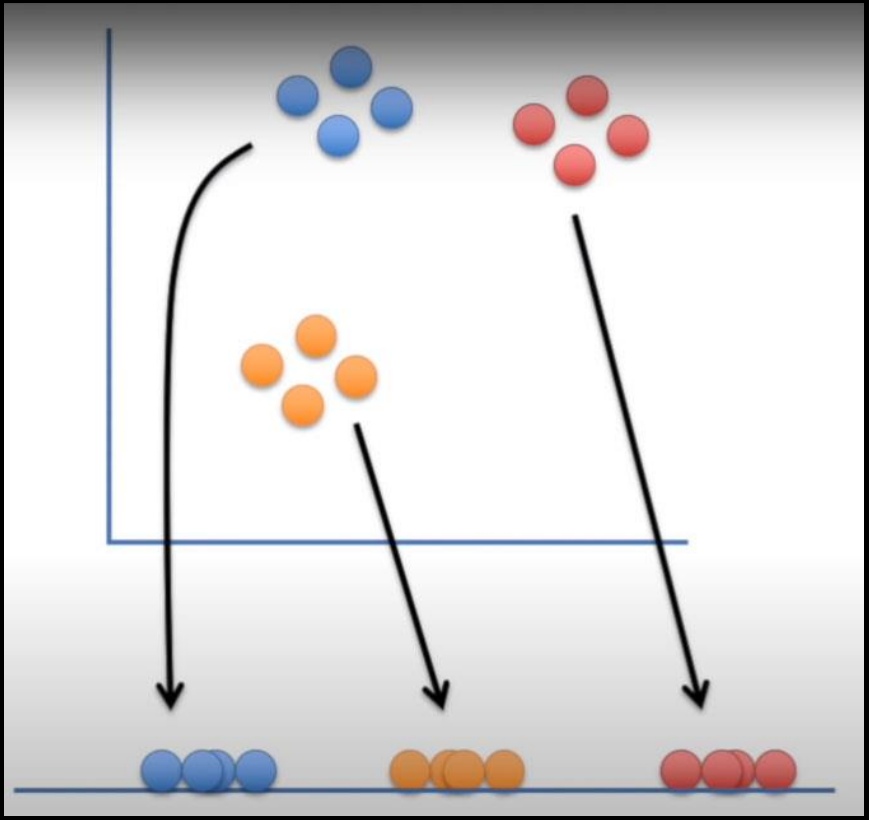


t-SNE and UMAP

- In Single Cell, primarily two methods are used.
  - t-SNE stands for “t-distributed stochastic neighbor embedding”.
  - UMAP stands for “Uniform Manifold Approximation and Projection”

- These are complex algorithms that we don’t have time to go into in any detail.

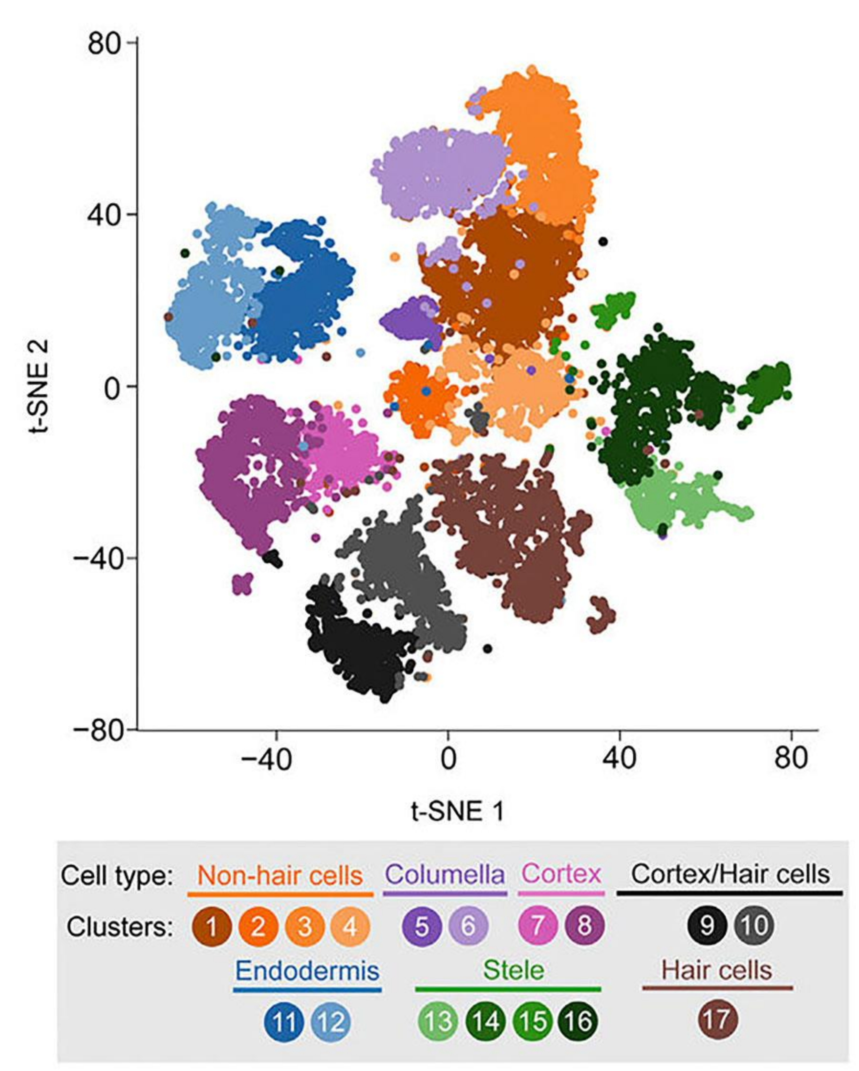
- In a nutshell they construct a probability distribution over the points in the high dimensional space and then try to approximate that in a lower dimensional space as closely as possible.



Single Cell t-SNE Example

- Single-Cell RNA-Seq of 12,198 Arabidopsis Root Cells Captures Diverse Cell Types.

- t-Distributed Stochastic Neighbor Embedding (t-SNE) dimensional reduction of 12,198 single Arabidopsis root cells. Cells were clustered into 17 populations


t-SNE 2
t-SNE 1
Cell type:
Clusters:
Non-hair cells
Columella
Cortex
Cortex/Hair cells
Endodermis
Stele
Hair cells