Dimension Reduction
PCA, tSNE, UMAP

v2020-11

Simon Andrews
simon.andrews@babraham.ac.uk

Babraham Bioinformatics

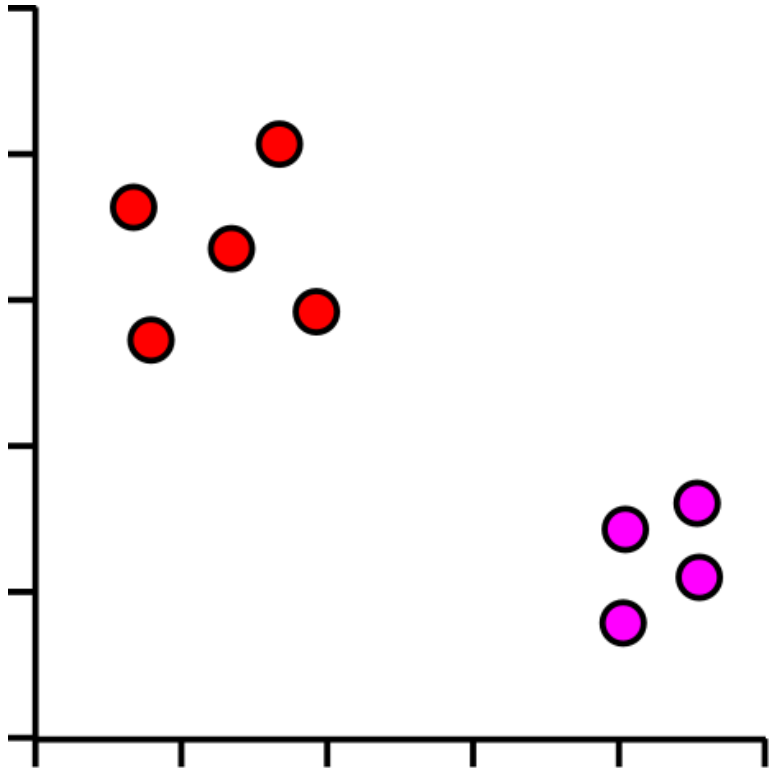
Where are we heading?

Gene	Description	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5
Inpp5d	inositol polyphosphate-5-phosphatase D	7.00	5.45	5.89	6.03	5.75
Aim2	absent in melanoma 2	3.01	4.37	4.59	4.38	4.18
Gldn	gliomedin	3.48	3.63	4.61	4.70	4.74
Frem2	Fras1 related extracellular matrix protein 2	4.75	4.66	3.46	3.74	3.45
Rps3a1	ribosomal protein S3A1	6.10	7.23	7.44	7.36	7.34
Slc38a3	solute carrier family 38, member 3	1.90	3.16	3.52	3.61	3.19
Mt1	metallothionein 1	5.07	6.49	6.46	6.04	6.05
C1s1	complement component 1, s subcomponent 1	2.74	3.02	3.86	4.10	4.10
Cds1	CDP-diacylglycerol synthase 1	4.55	4.22	3.80	3.16	3.12
Ifi44	interferon-induced protein 44	4.82	4.52	3.87	3.42	3.59
Lefty2	left-right determination factor 2	6.95	6.28	5.88	5.60	5.61
Fmr1nb	fragile X mental retardation 1 neighbor	4.28	2.78	3.10	3.25	2.57
Tagln	transgelin	7.93	7.91	7.20	7.02	6.68

Each dot is a cell

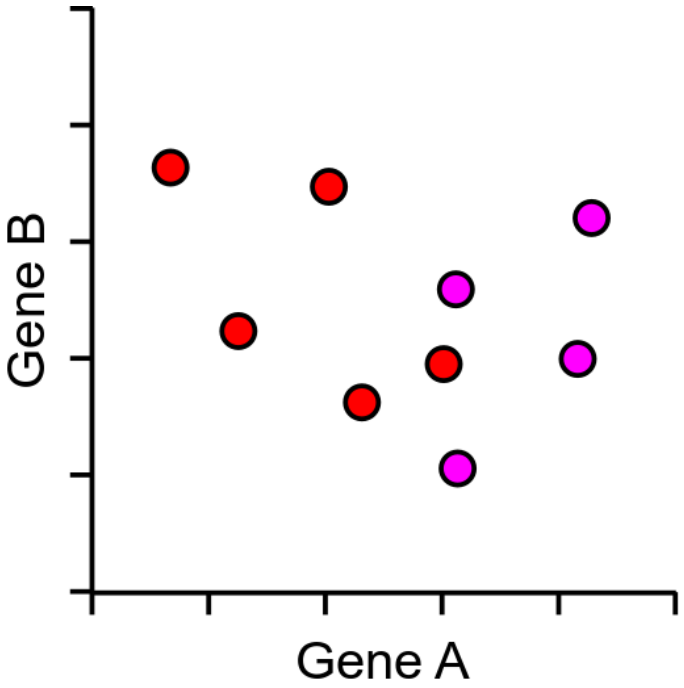
Groups of dots are similar cells

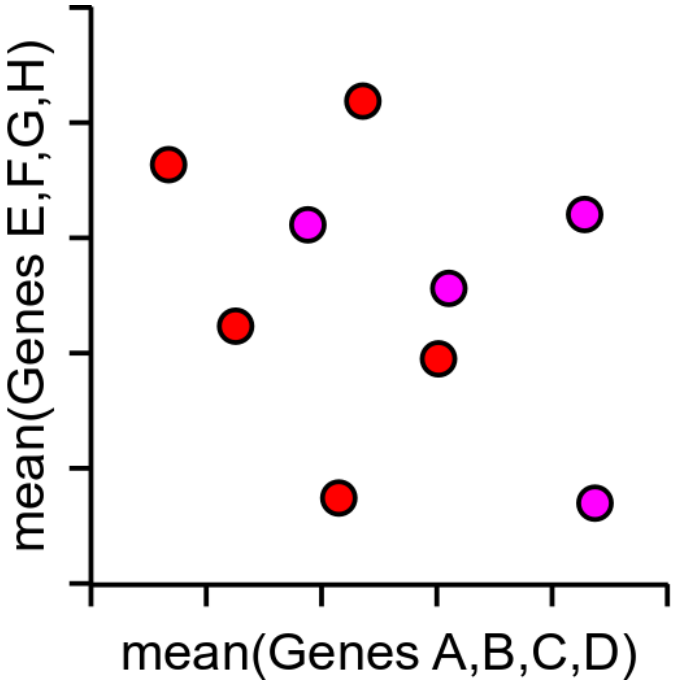
Separation of groups could be interesting biology

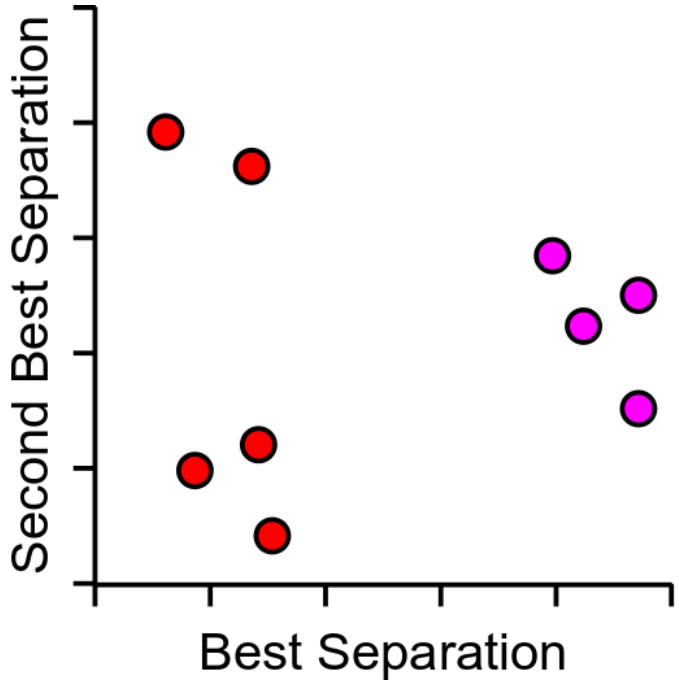


Too much data!

- 5000 cells and 2500 measured genes
- Realistically only 2 dimensions we can plot (x,y)







Principle Components Analysis

- Method to optimally summarise large multi-dimensional datasets
- Can find a smaller number of dimensions (ideally 2) which retain most of the useful information in the data

- Builds a recipe for converting large amounts of data into a single value, called a Principle Component (PC), eg:

PC = (GeneA*10)+(GeneB*3)+(GeneC*-4)+(GeneD*-20)…

Principle Components Analysis

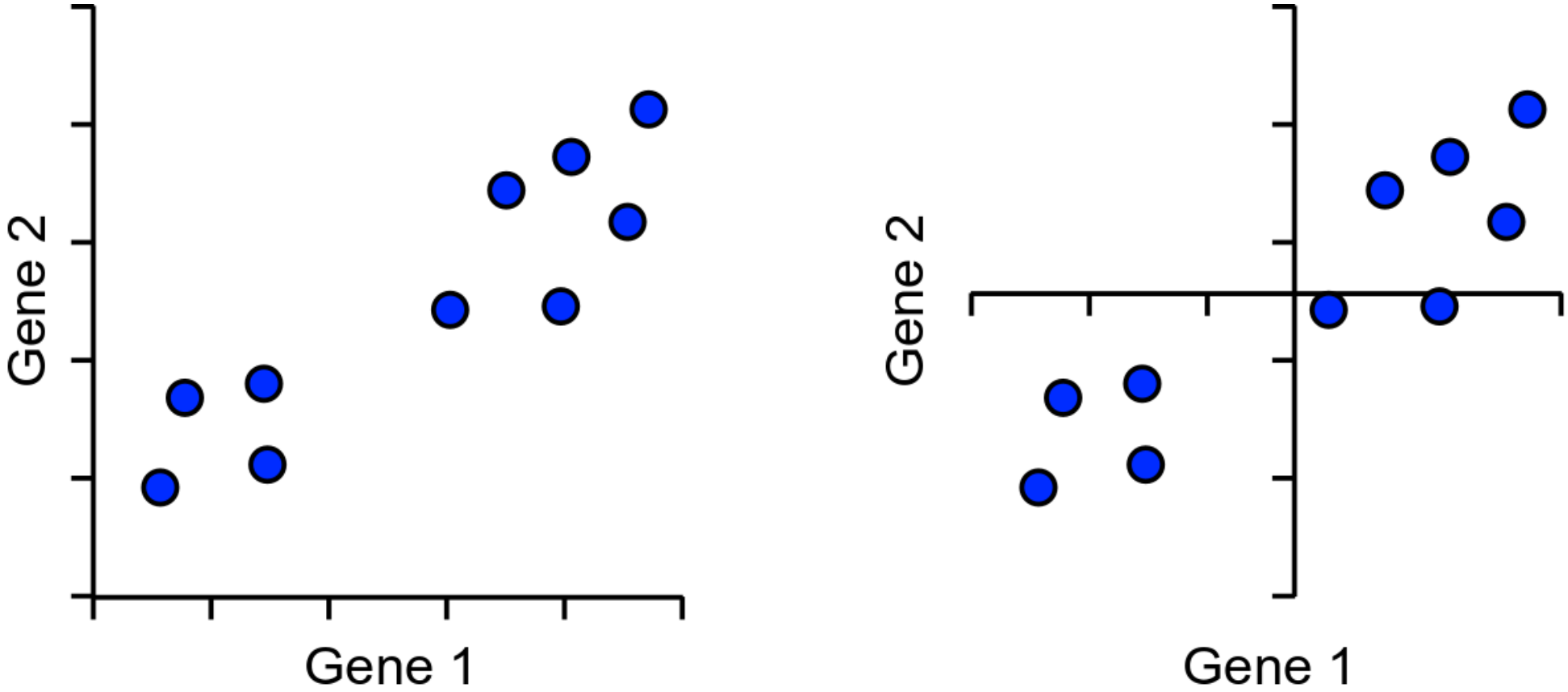
- Method to optimally summarise large multi-dimensional datasets
- Can find a smaller number of dimensions (ideally 2) which retain most of the useful information in the data

- Builds a recipe for converting large amounts of data into a single value, called a Principle Component (PC), eg:

PC = (GeneA*10)+(GeneB*3)+(GeneC*-4)+(GeneD*-20)…

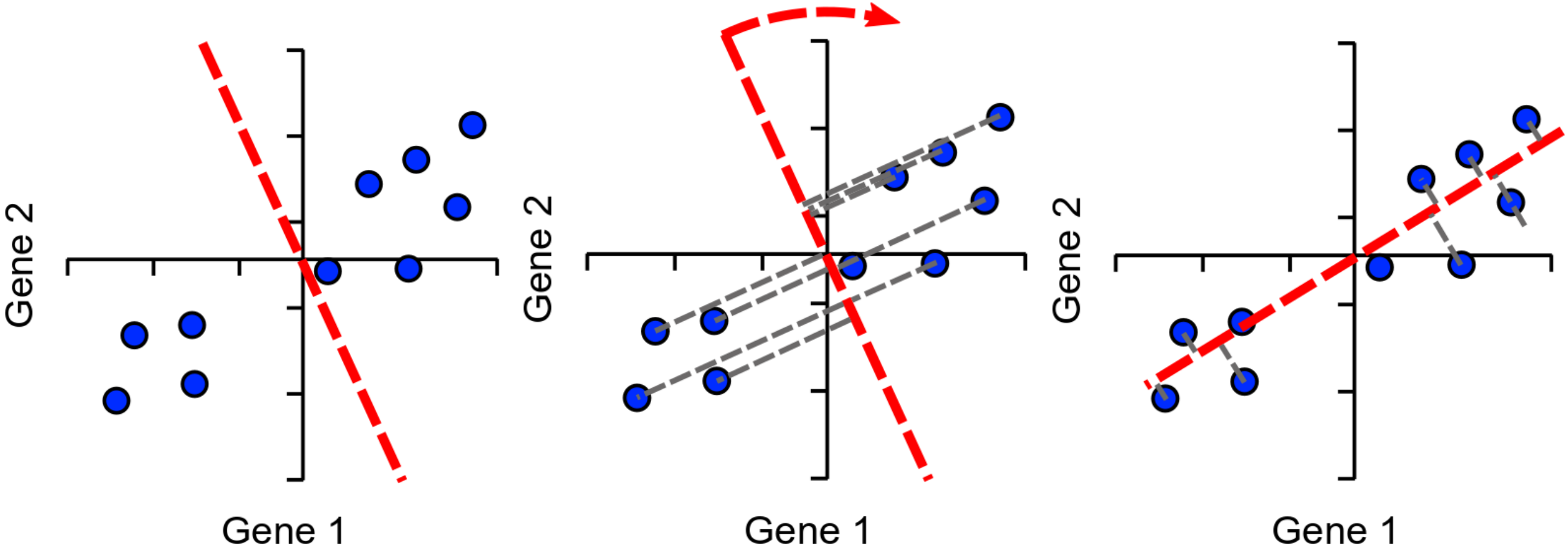
How does PCA work?

- Simple example using 2 genes and 10 cells

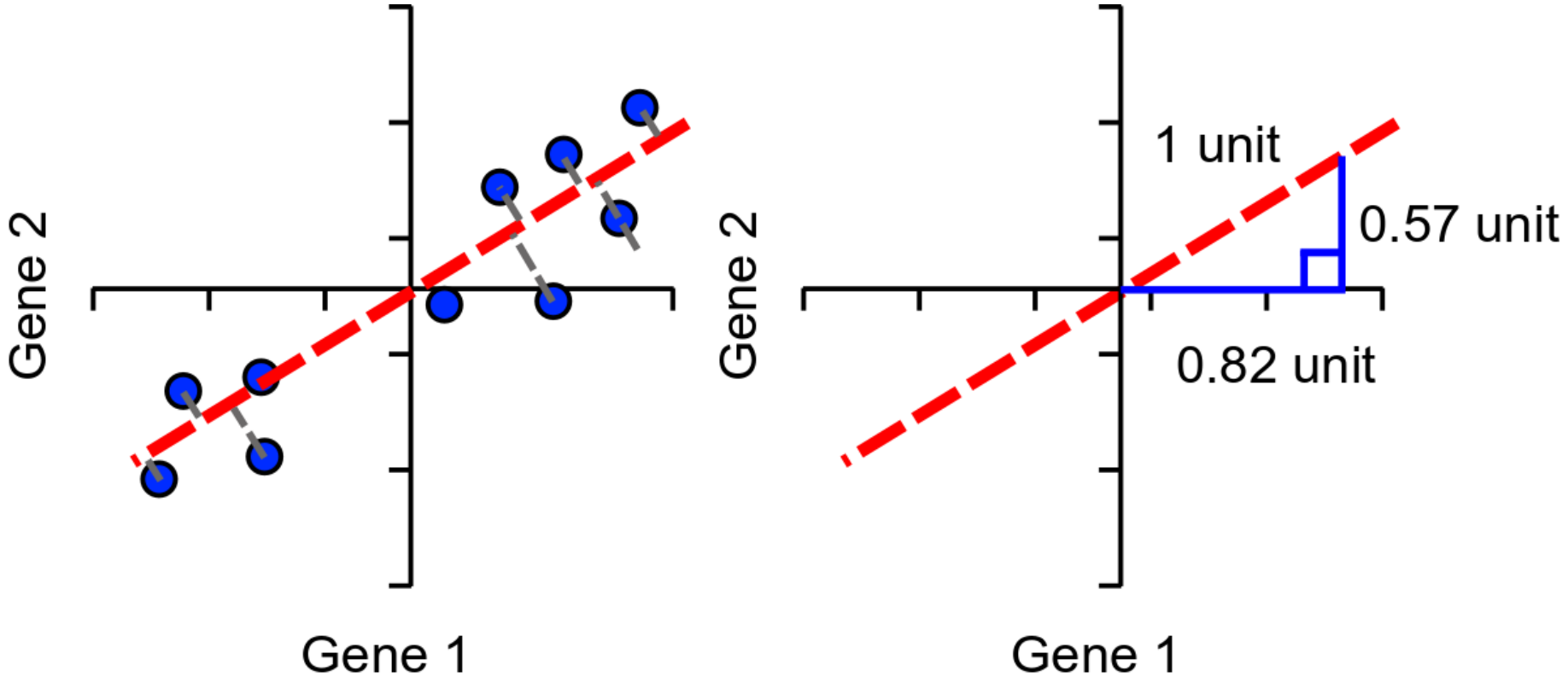


How does PCA work?

- Find line of best fit, passing through the origin



Assigning Loadings to Genes



Single Vector or '**eigenvector**'

Loadings:

- Gene1 = 0.82
- Gene2 = 0.57

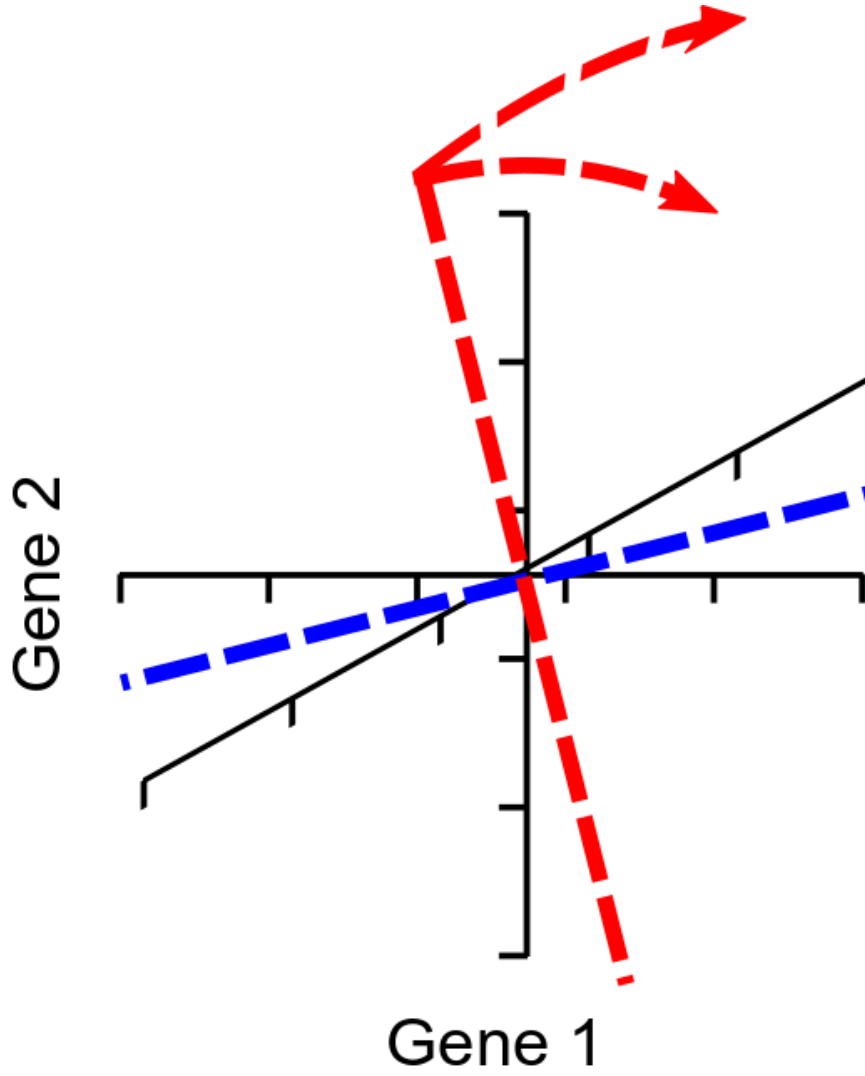
Higher loading equals more influence on PC

More dimensions

- The same idea extends to larger numbers of dimensions (n)

- Calculation of first PC rotates in (n-1) -dimensions
  - Next PC is perpendicular to PC2, but rotated similarly (n-2)
  - Last PC is remaining perpendicular (no choice)
  - Same number of PCs as genes

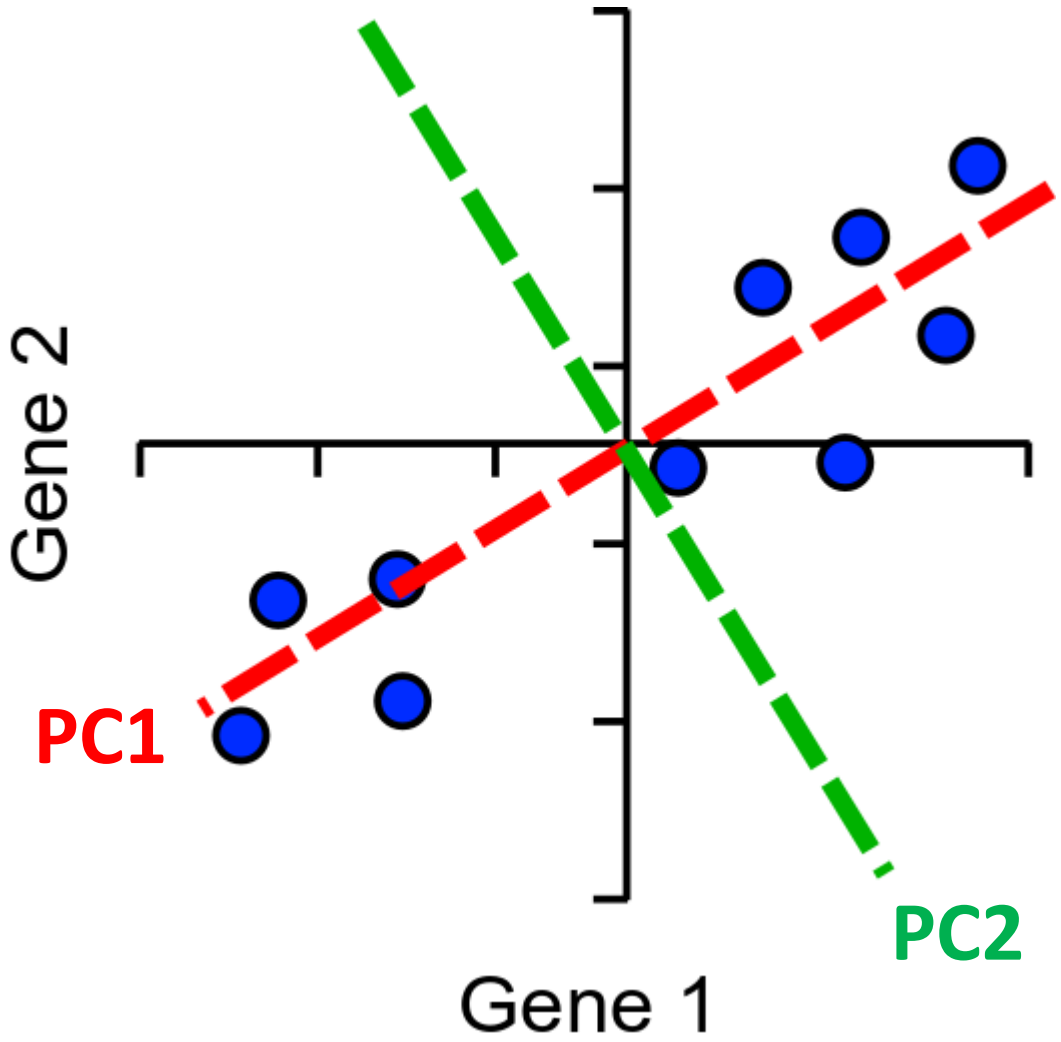
Example with 3 dimensions



Explaining Variance

- Each PC always explains some proportion of the total variance in the data. Between them they explain everything
  - PC1 always explains the most
  - PC2 is the next highest etc. etc.
- Since we only plot 2 dimensions we'd like to know that these are a good explanation
- How do we calculate this?

Explaining variance

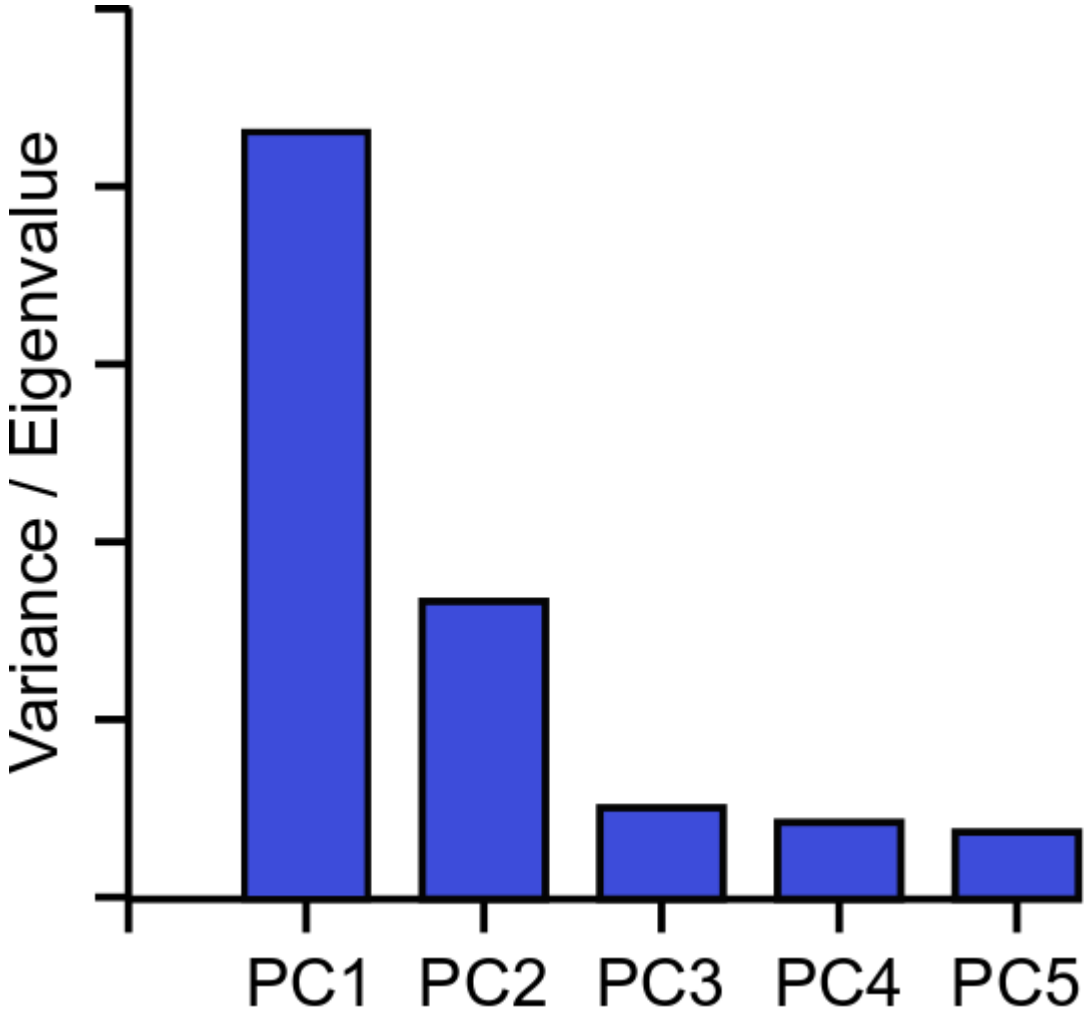


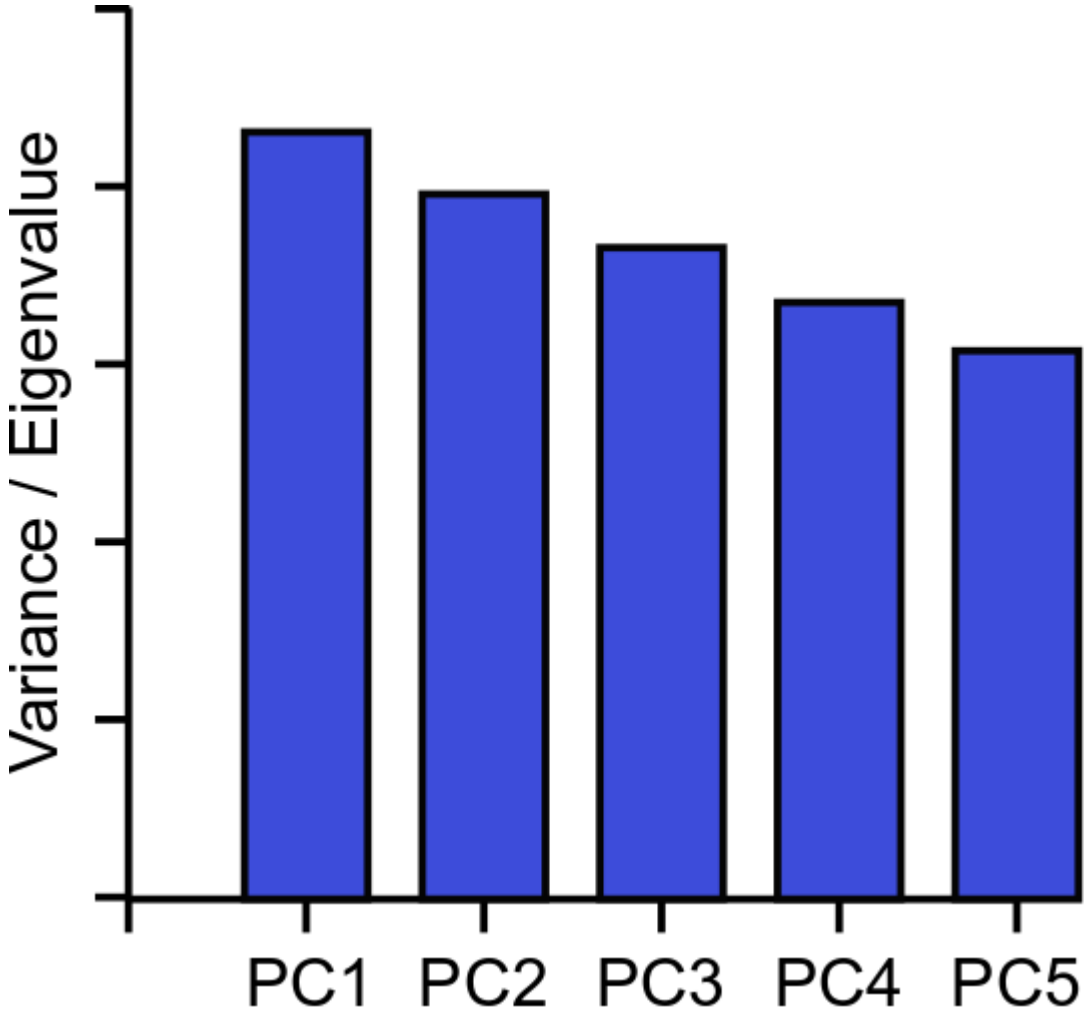
- Project onto PC
- Calculate distance to the origin

- Calculate sum of squared differences (SSD)
  - This is a measure of variance called the ‘eigenvalue’

  - Divide by (points-1) to get actual variance

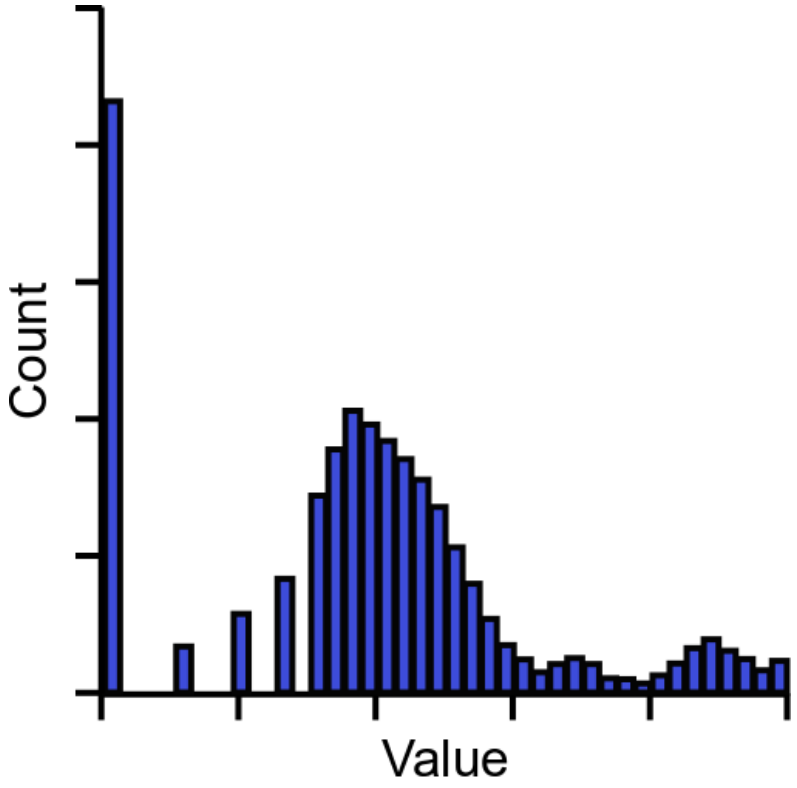
Explaining Variance – Scree Plots



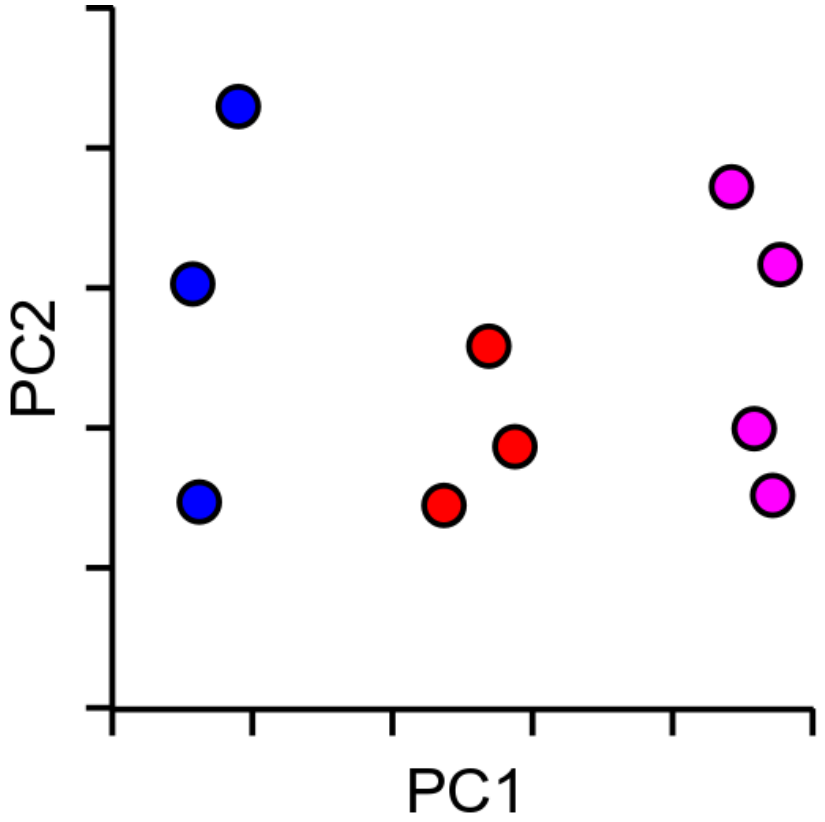


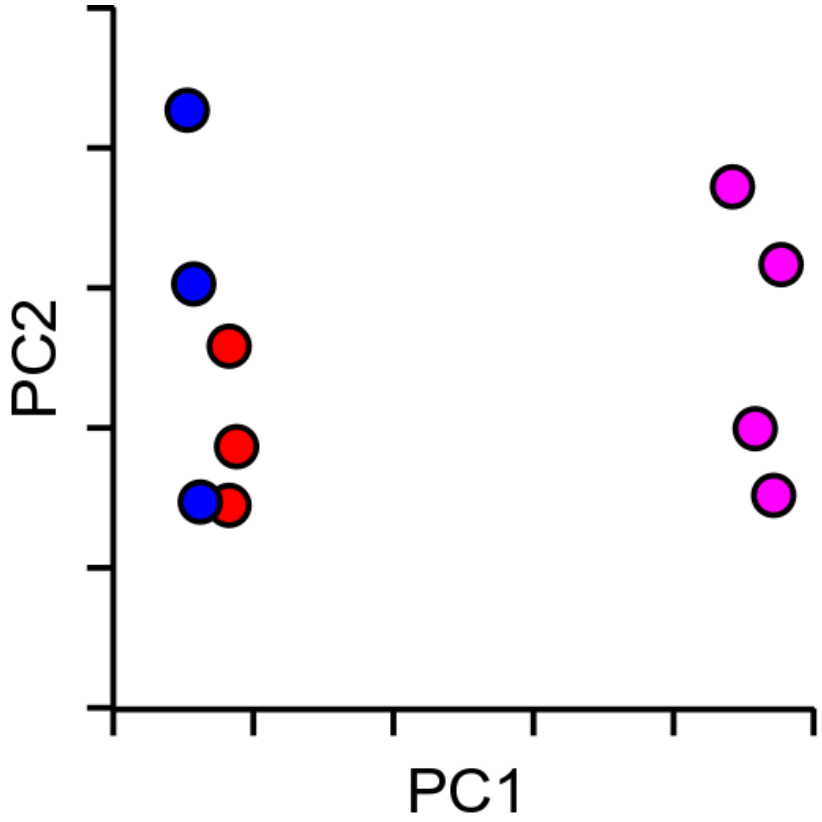
So PCA is great then?

- Kind of…



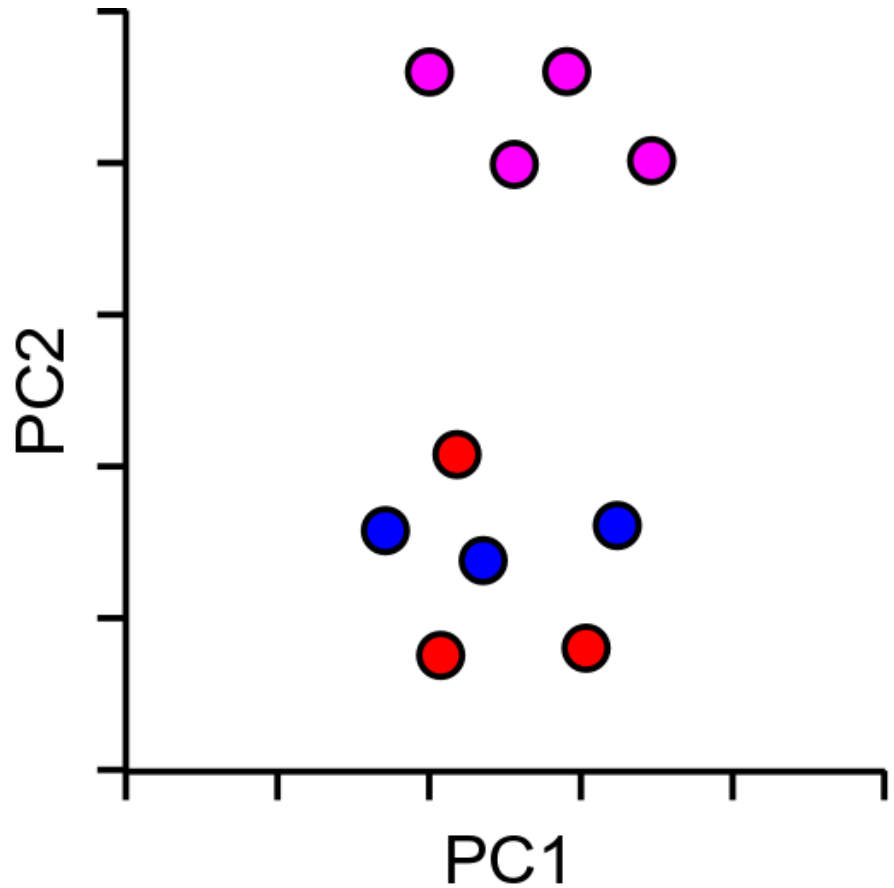
Non-linear separation of values

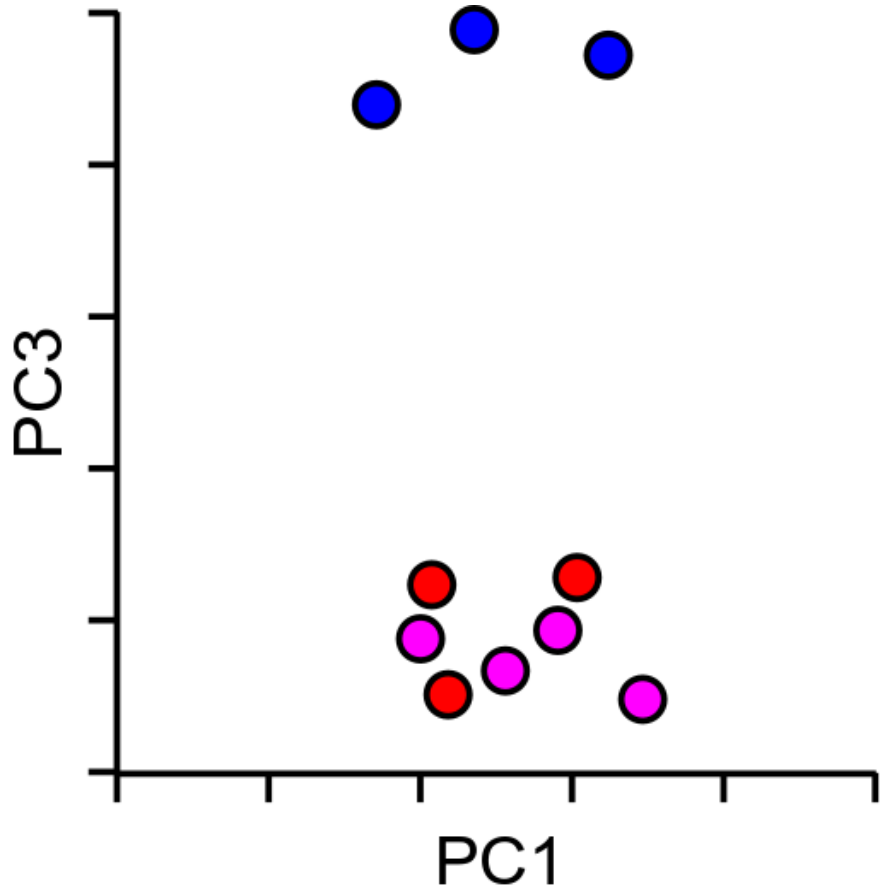




So PCA is great then?

- Kind of…





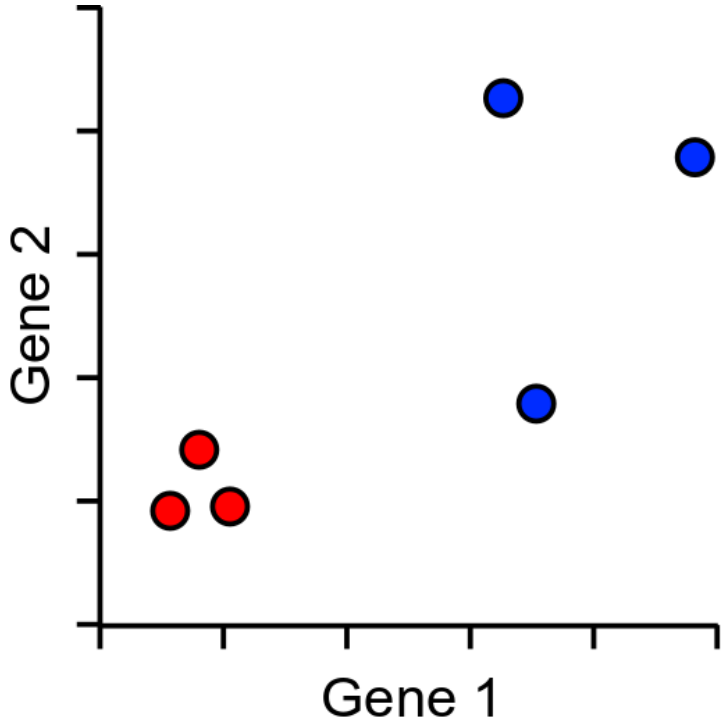
Not optimised for 2-dimensions

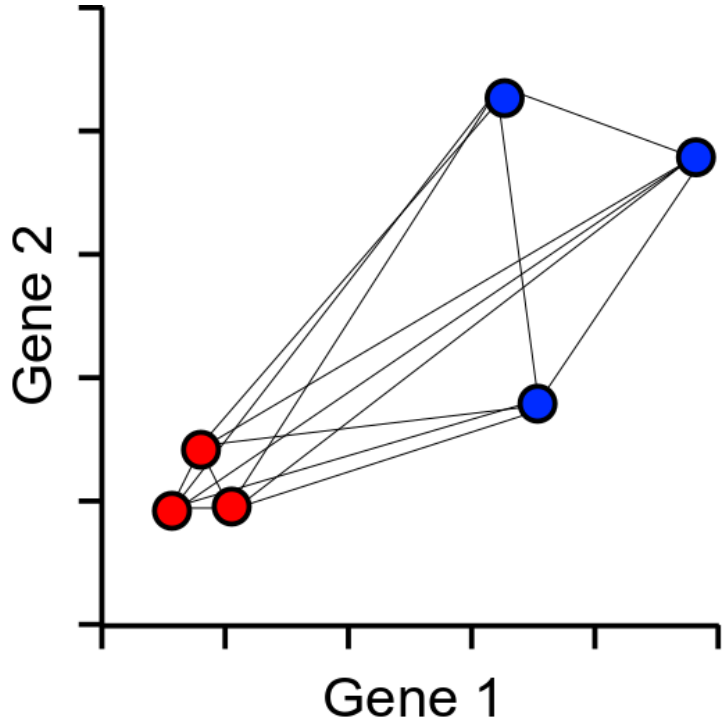
tSNE to the rescue…

- T-Distributed Stochastic Neighbour Embedding
- Aims to solve the problems of PCA
  - Non-linear scaling to represent changes at different levels
  - Optimal separation in 2-dimensions

How does tSNE work?

- Based around all-vs-all table of pairwise cell to cell distances

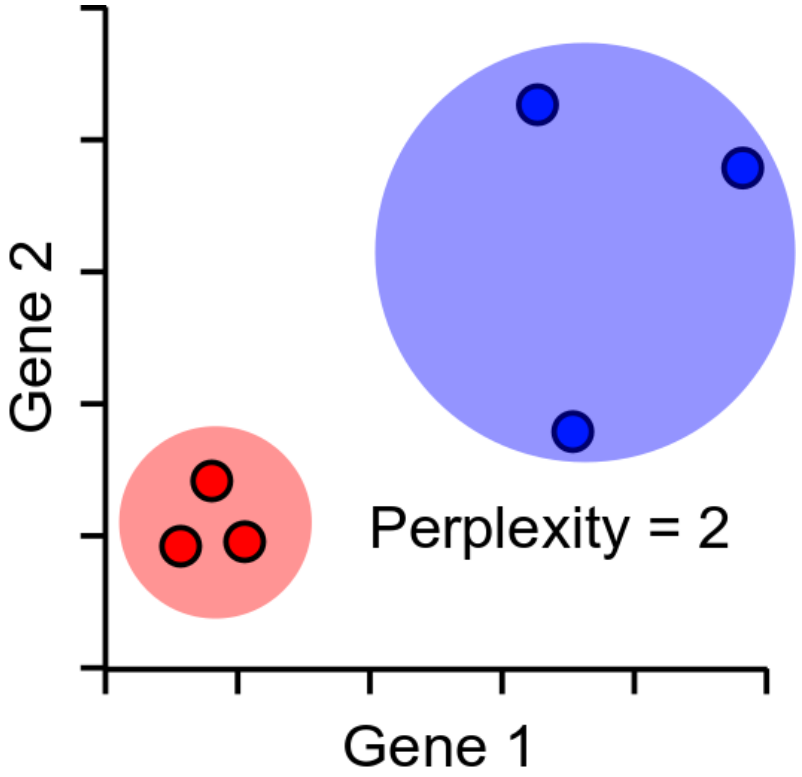




	0	10	10	295	158	153
	9	0	1	217	227	213
	1	8	0	154	225	238
	205	189	260	0	23	45
	248	227	246	44	0	54
	233	176	184	41	36	0

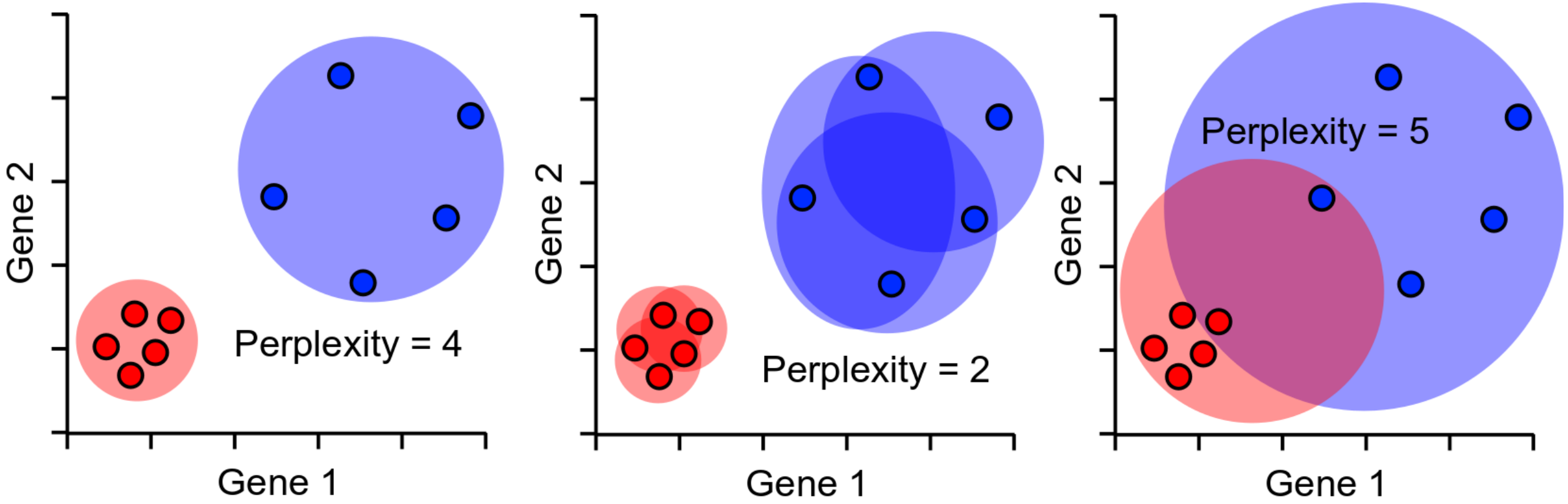
Distance scaling and perplexity

- Perplexity = expected number of neighbours within a cluster
- Distances scaled relative to perplexity neighbours



	0	4	6	586	657	836
	4	0	4	815	527	776
	9	3	0	752	656	732
	31	28	29	0	4	7
	31	24	25	4	0	7
	40	37	32	8	8	0

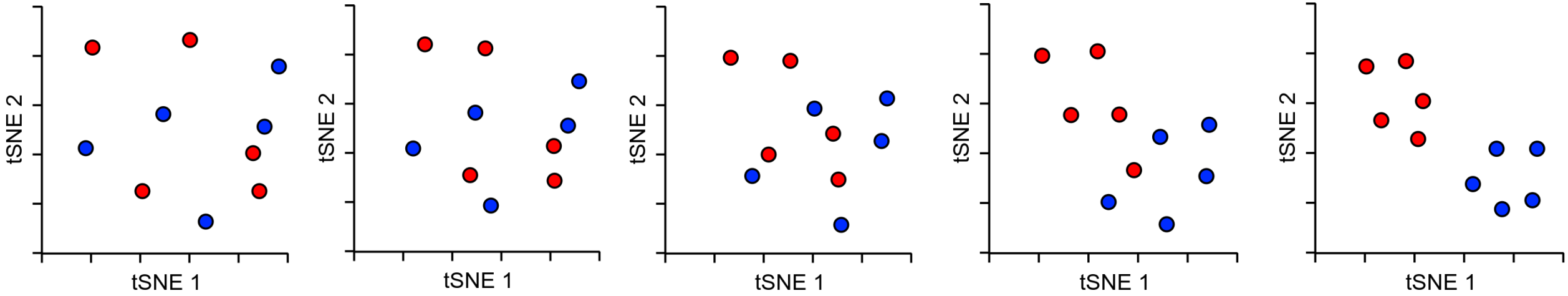
Perplexity Robustness



tSNE Projection

- Normally 2D, but can be any number of dimensions
- Randomly scatter all points within the space
- Start a simulation
  - Aim is to make the point distances match the distance matrix
  - Shuffle points based on how well they match
  - Stop after fixed number of iterations, or
  - Stop after distances have converged

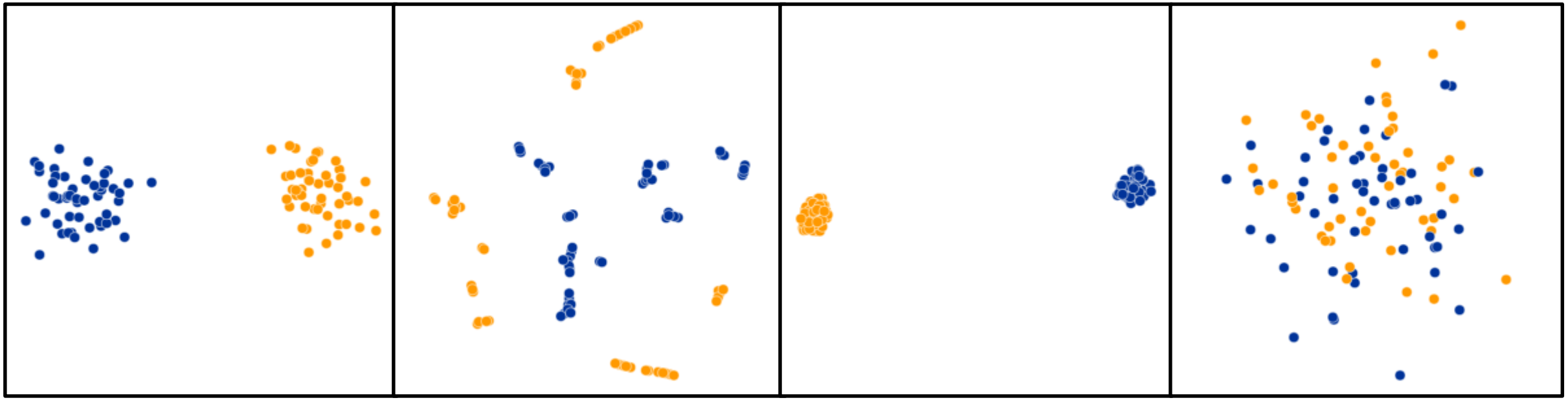
tSNE Projection



- X and Y don’t mean anything (unlike PCA)
- Distance doesn’t mean anything (unlike PCA)
- Close proximity is highly informative
- Distant proximity isn’t very interesting
- Can’t rationalise distances, or add in more data

tSNE Practical Examples

Perplexity Settings Matter

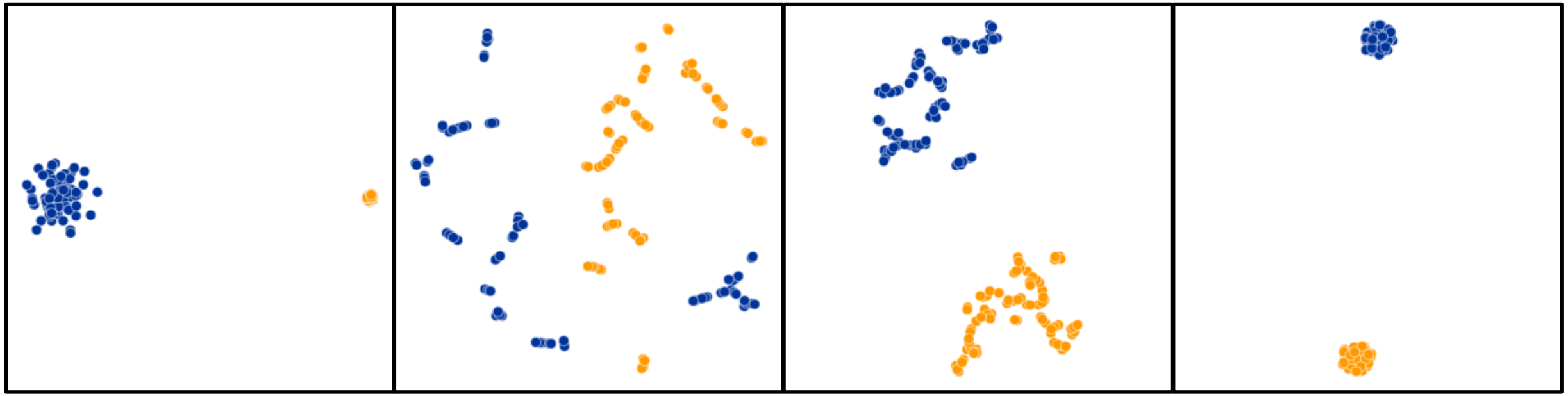


Original | Perplexity = 2 | Perplexity = 30 | Perplexity = 100

https://distill.pub/2016/misread-tsne/

tSNE Practical Examples

Cluster Sizes are Meaningless

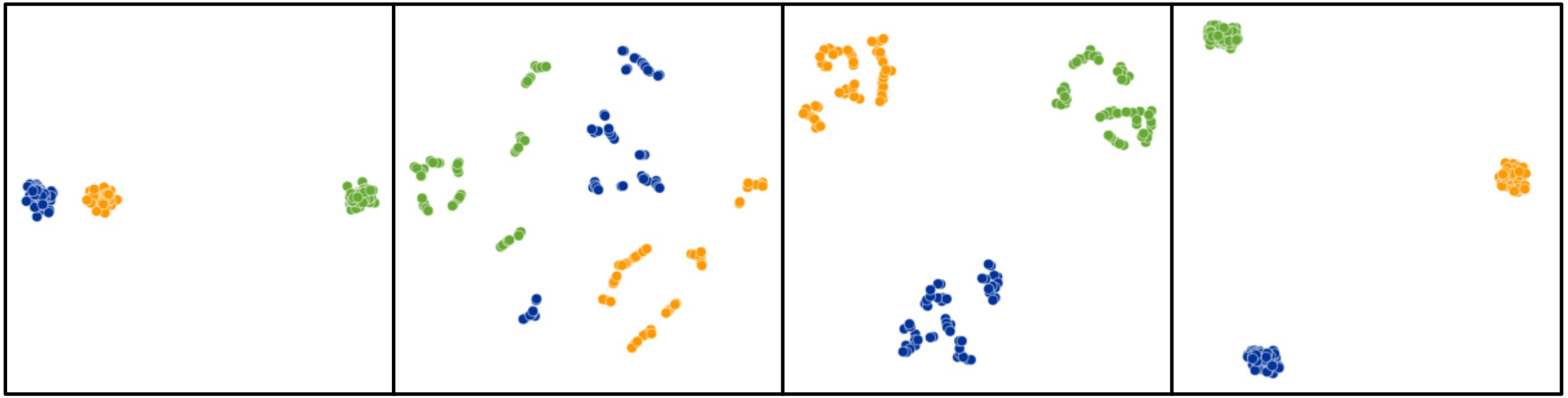


Original | Perplexity = 2 | Perplexity = 5 | Perplexity = 50

https://distill.pub/2016/misread-tsne/

tSNE Practical Examples

Distances between clusters can’t be trusted

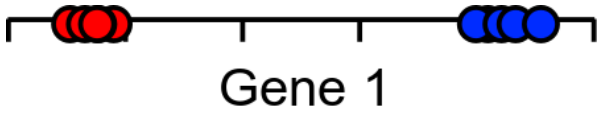


Original | Perplexity = 2 | Perplexity = 5 | Perplexity = 30

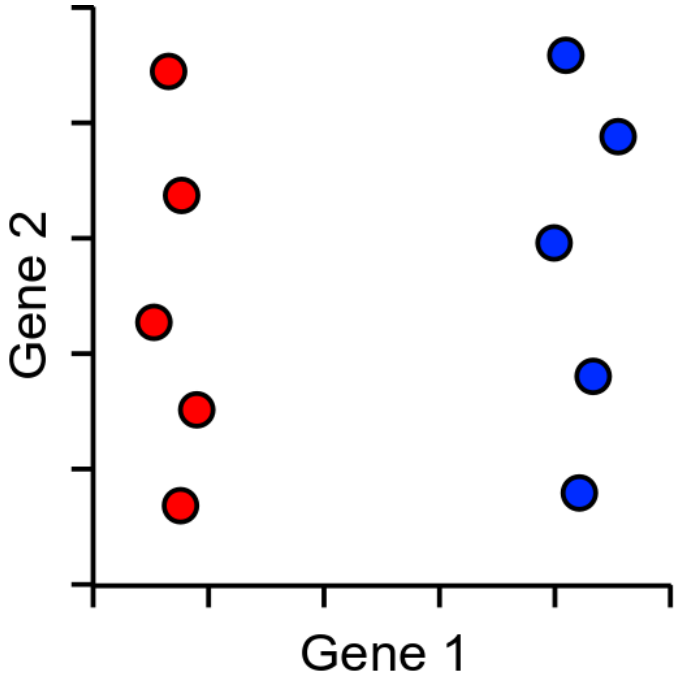
https://distill.pub/2016/misread-tsne/

So tSNE is great then?

- Kind of…
- Imagine a dataset with only one super informative gene



Distance within cluster = low
Distance between clusters = high



Distance within cluster = higher
Distance between clusters = lower

- Now 3 genes
- Now 3,000 genes
- Everything is the same distance from everything

So everything sucks?

- PCA
  - Requires more than 2 dimensions
  - Thrown off by quantised data
  - Expects linear relationships
- tSNE
  - Can't cope with noisy data
  - Loses the ability to cluster

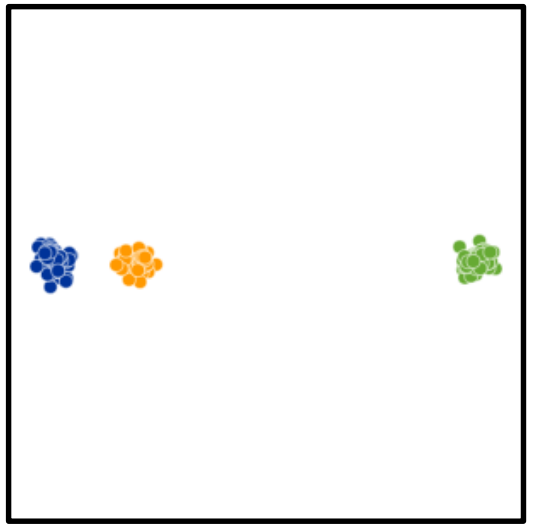
Answer: Combine the two methods, get the best of both worlds

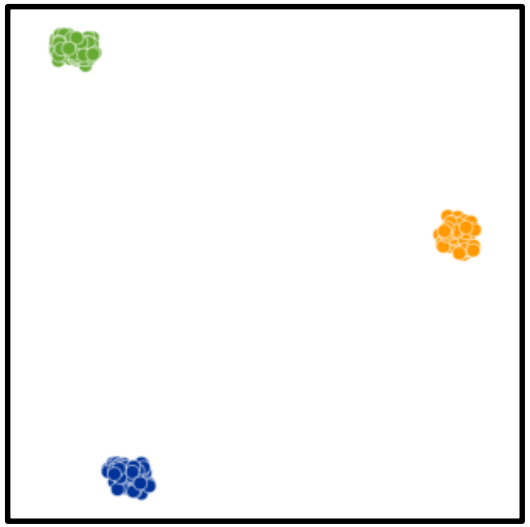
- PCA
  - Good at extracting signal from noise
  - Extracts informative dimensions
- tSNE
  - Can reduce to 2D well
  - Can cope with non-linear scaling

This is what CellRanger does in its default analysis

So PCA + tSNE is great then?

- Kind of…
  - tSNE is slow. This is probably it's biggest crime
    - tSNE doesn't scale well to large numbers of cells (10k+)
  - tSNE only gives reliable information on the closest neighbours large distance information is almost irrelevant





UMAP to the rescue!

- UMAP is a replacement for tSNE to fulfil the same role
- Conceptually very similar to tSNE, but with a couple of relevant (and somewhat technical) changes
- Practical outcome is:
  - UMAP is quite a bit quicker than tSNE
  - UMAP can preserve more global structure than tSNE*
  - UMAP can run on raw data without PCA preprocessing*
  - UMAP can allow new data to be added to an existing projection

* In theory, but possibly not in practice

UMAP differences

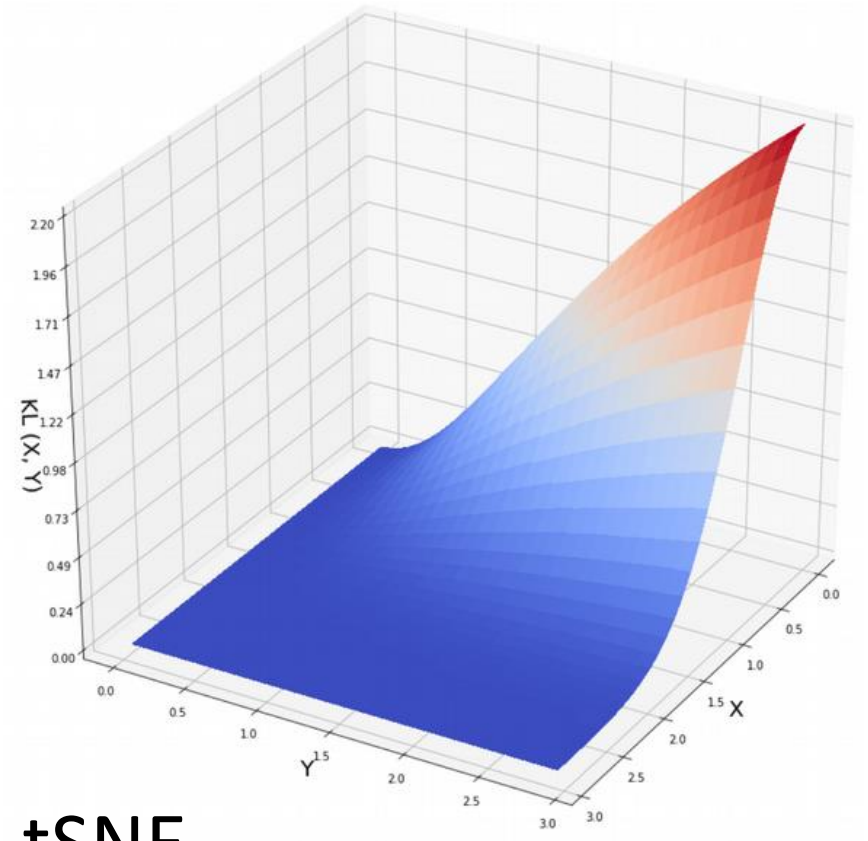
- Instead of the single perplexity value in tSNE, UMAP defines
  - **Nearest neighbours**: the number of expected nearest neighbours – basically the same concept as perplexity
  - **Minimum distance**: how tightly UMAP packs points which are close together
- Nearest neighbours will affect the influence given to global vs local information. Min dist will affect how compactly packed the local parts of the plot are.

UMAP differences

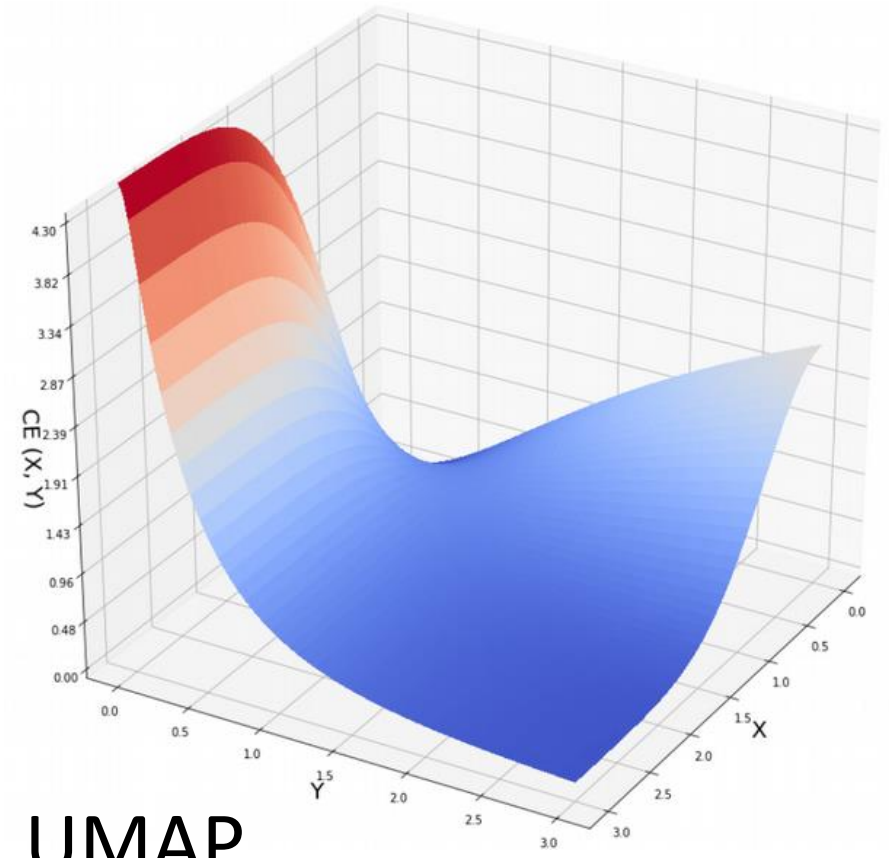
- Speed – mostly a level of maths I'm not going to get into!
  - UMAP skips a normalisation step in the calculation of high dimensional distances which speeds it up
  - In the 2D projection UMAP uses a more efficient method to shuffle the cells into their final position
    - Doesn't have to measure every cell to decide on what to move
    - Uses an algorithm which can be multi-threaded
    - Algorithm is more deterministic, allowing more data to be projected later

UMAP differences

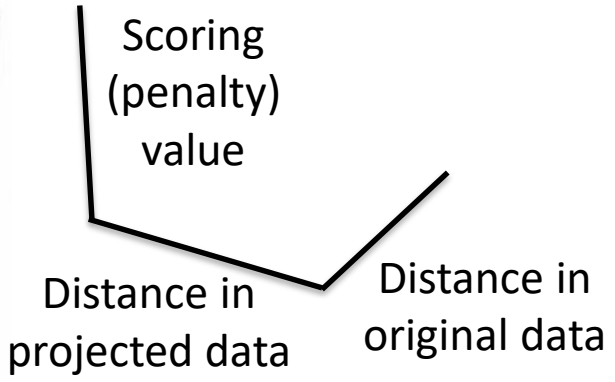
- Structure preservation – mostly in the 2D projection scoring



tSNE



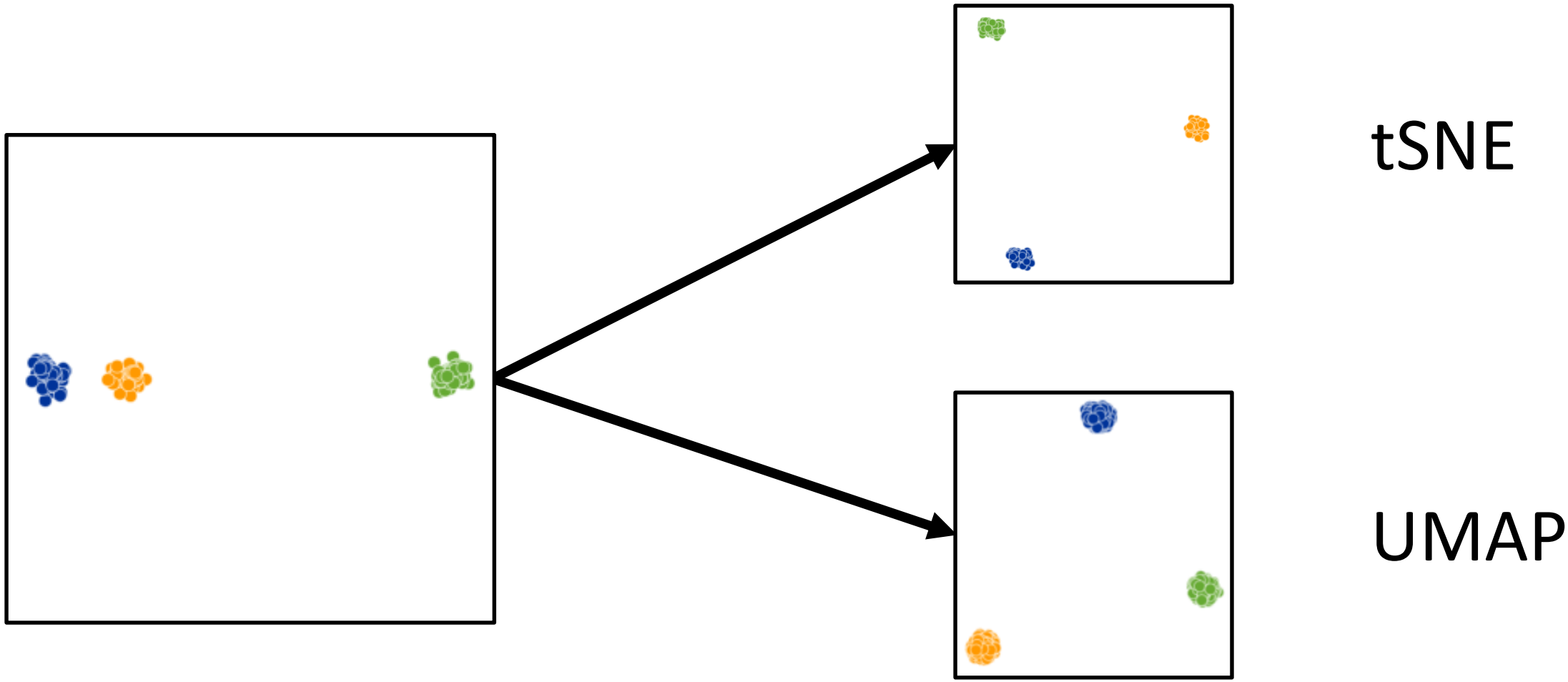
UMAP



https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668

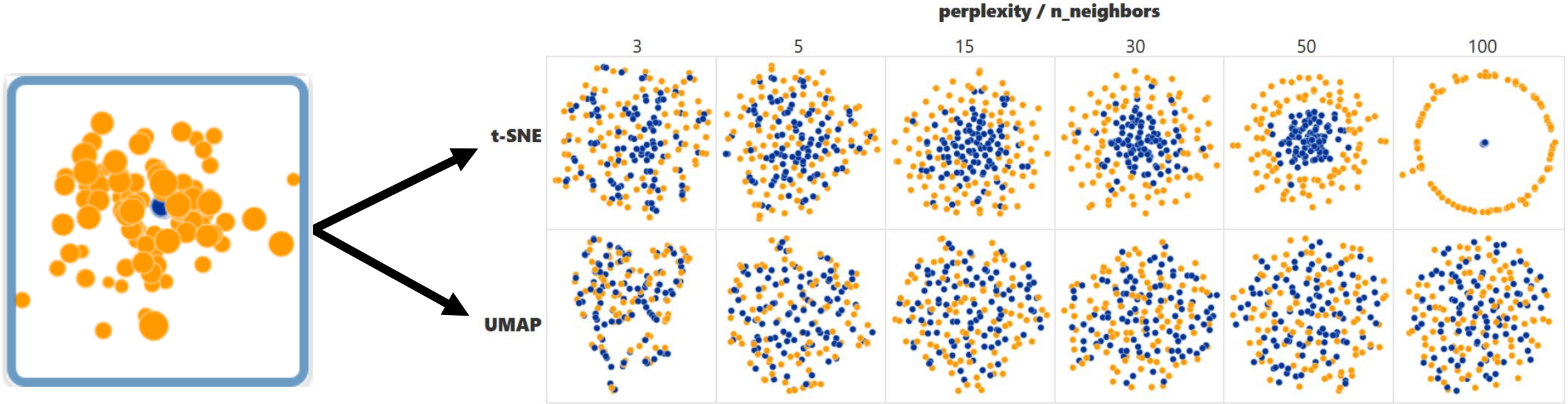
So UMAP is great then?

- Kind of…



So UMAP is great then?

- Kind of…

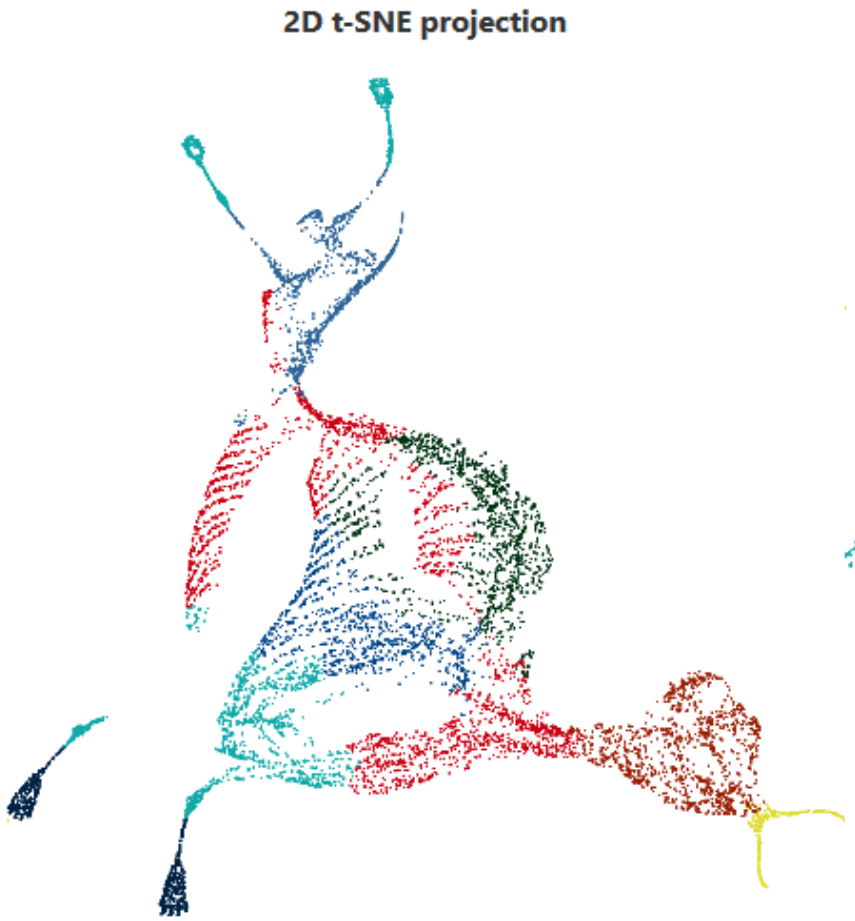


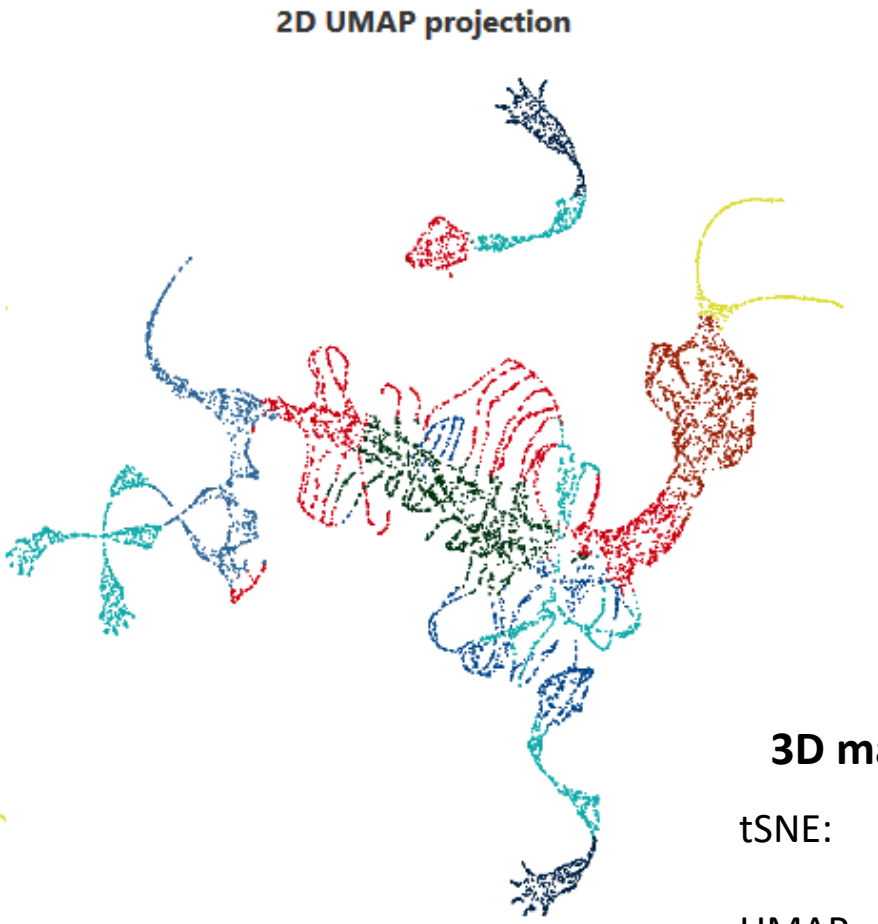
- It may perform better on more complex datasets
- It’s certainly quicker

https://pair-code.github.io/understanding-umap/

So UMAP is all hype then?

- No, it really does better for some datasets…





3D mammoth skeleton projected into 2D

tSNE:	Perplexity 2000	2h 5min
UMAP:	Nneigh 200, mindist 0.25,	3min

https://pair-code.github.io/understanding-umap/

Practical approach PCA + tSNE/UMAP

- Filter heavily before starting
  - Nicely behaving cells
  - Expressed genes
  - Variable genes
- Do PCA
  - Extract most interesting signal
  - Take top PCs. Reduce dimensionality (but not to 2)
- Do tSNE/UMAP
  - Calculate distances from PCA projections
  - Scale distances and project into 2-dimensions