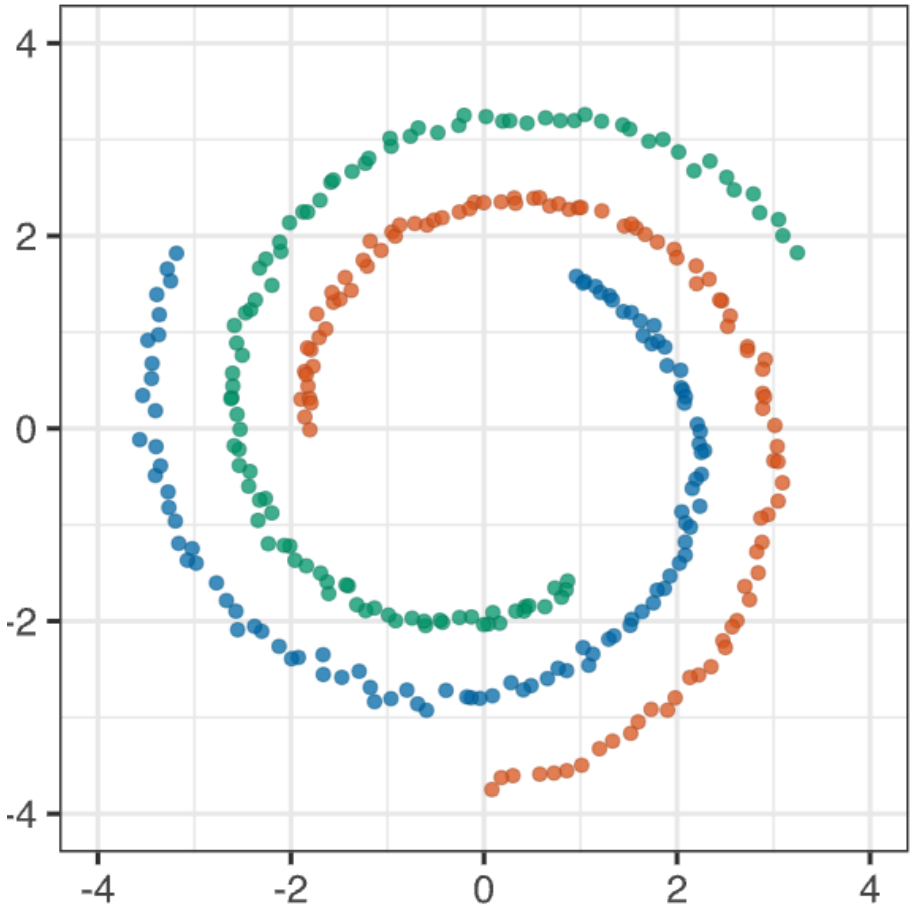
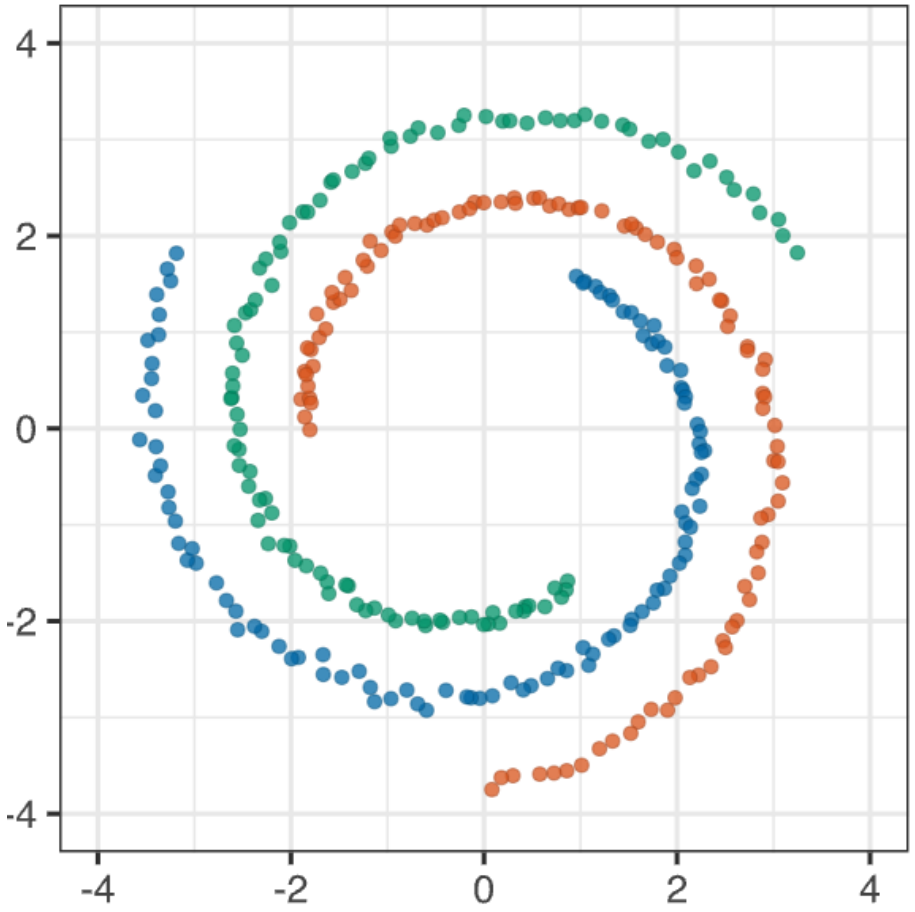
CSCI 491: Data Visualization

16- Dimension Reduction 2

Which direction is PC_1?



What if a rotation cannot disentangle the data?



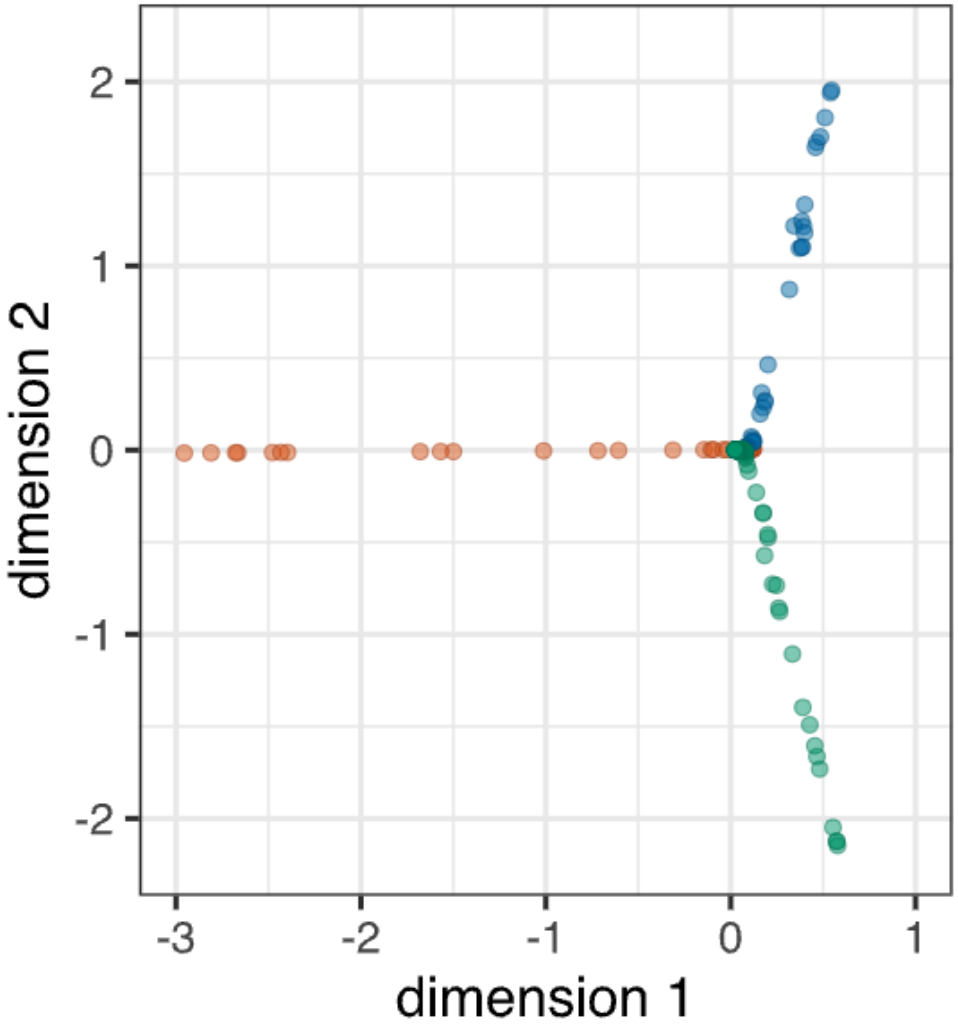
PCA analysis of intertwined spirals is not useful

What are the alternatives?

- Kernel PCA performs PCA in a hypothetical, higher-dimensional space
- With more dimensions, data points become more separable
- Importantly, the space is never explicitly constructed (kernel trick)
- Results from kernel PCA depend on choice of kernel

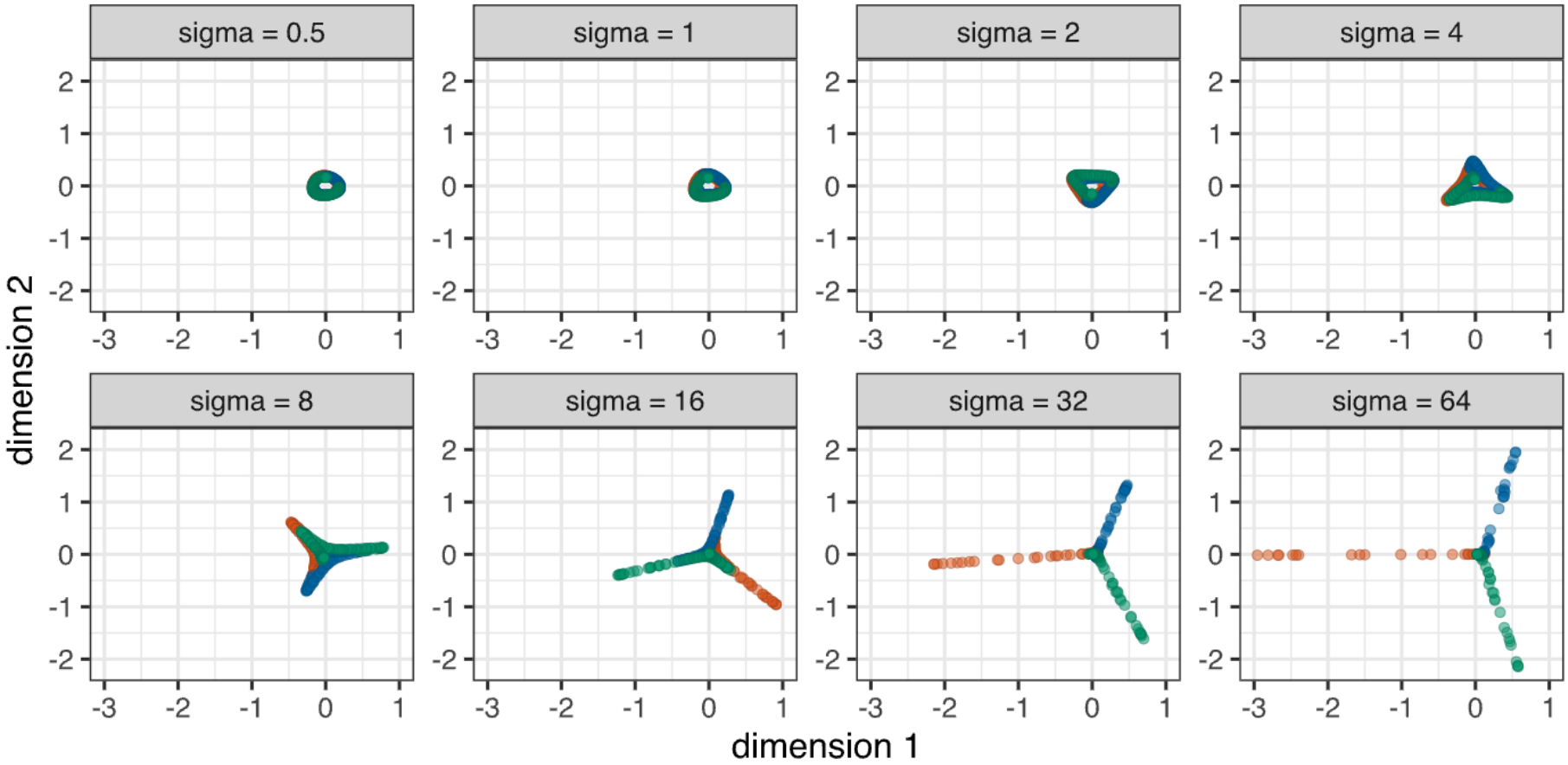
https://www.baeldung.com/cs/kernel-principal-component-analysis

Kernel PCA can separate the spirals



Gaussian kernel, sigma = 64

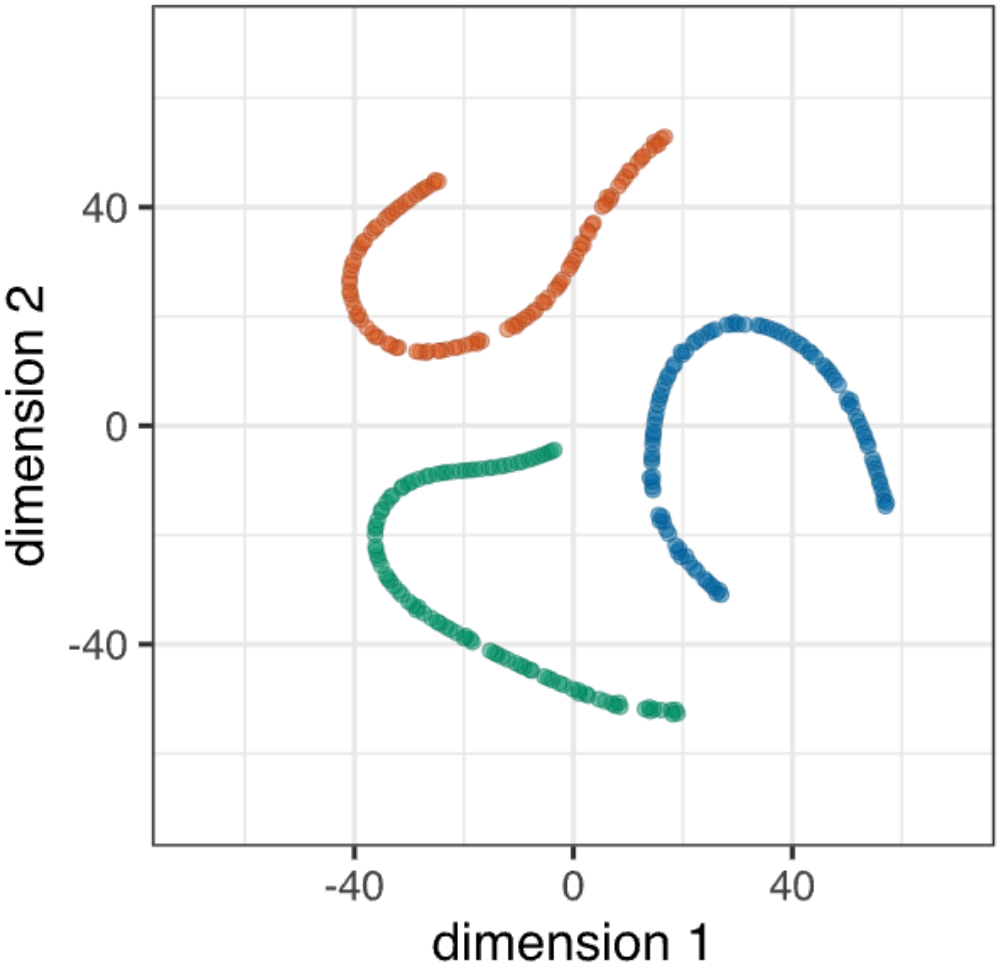
But we need to choose the right sigma value



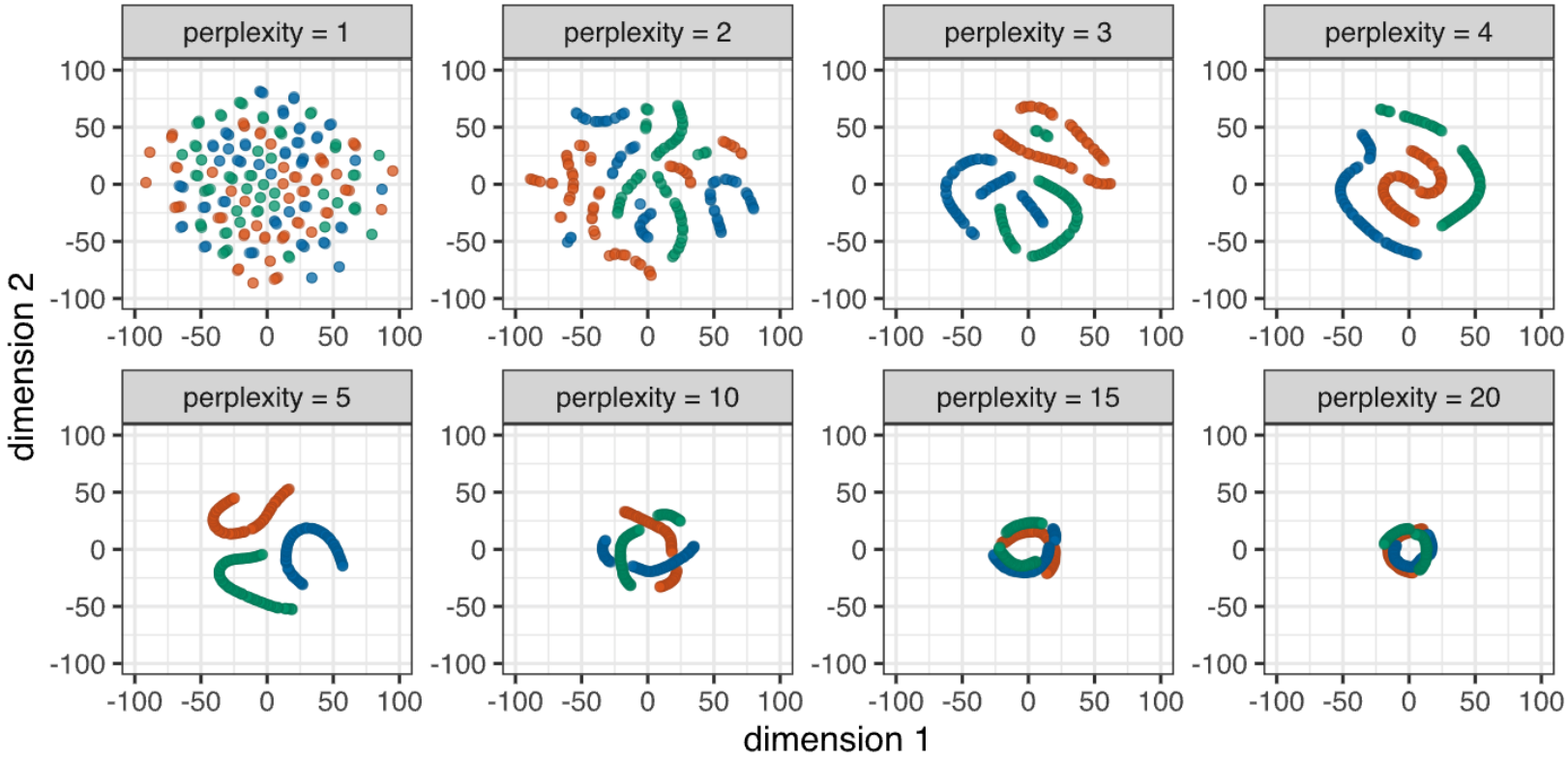
Other approaches

- t-SNE: t-distributed stochastic neighbor embedding
- UMAP: Uniform manifold approximation and projection
- Both algorithms look at the local distances between points in the original data space and try to reproduce them in the low-dimensional representation
- Both methods are stochastic, meaning they incorporate elements of randomness and can produce different results when run multiple times.

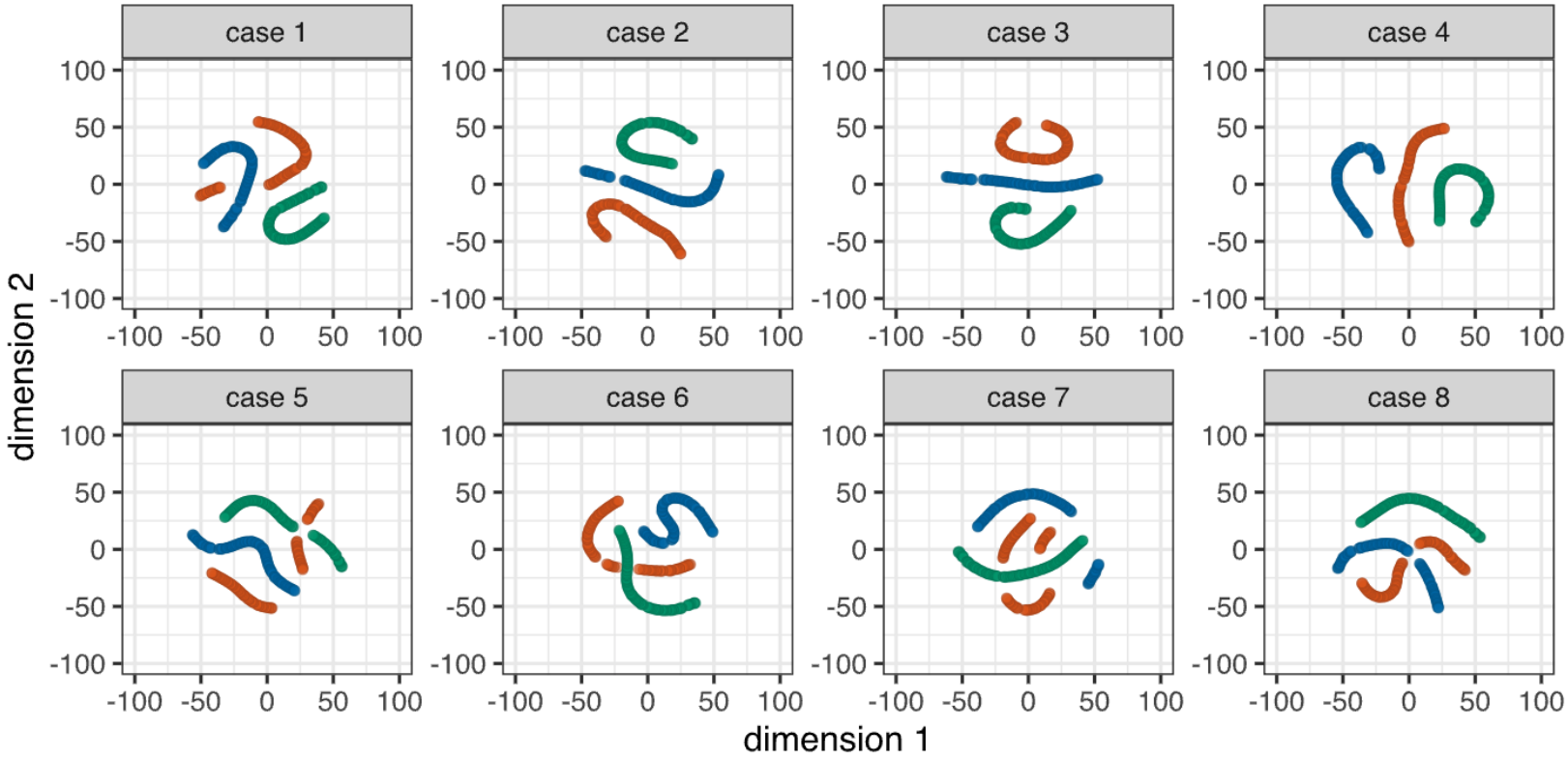
t-SNE can separate the spirals



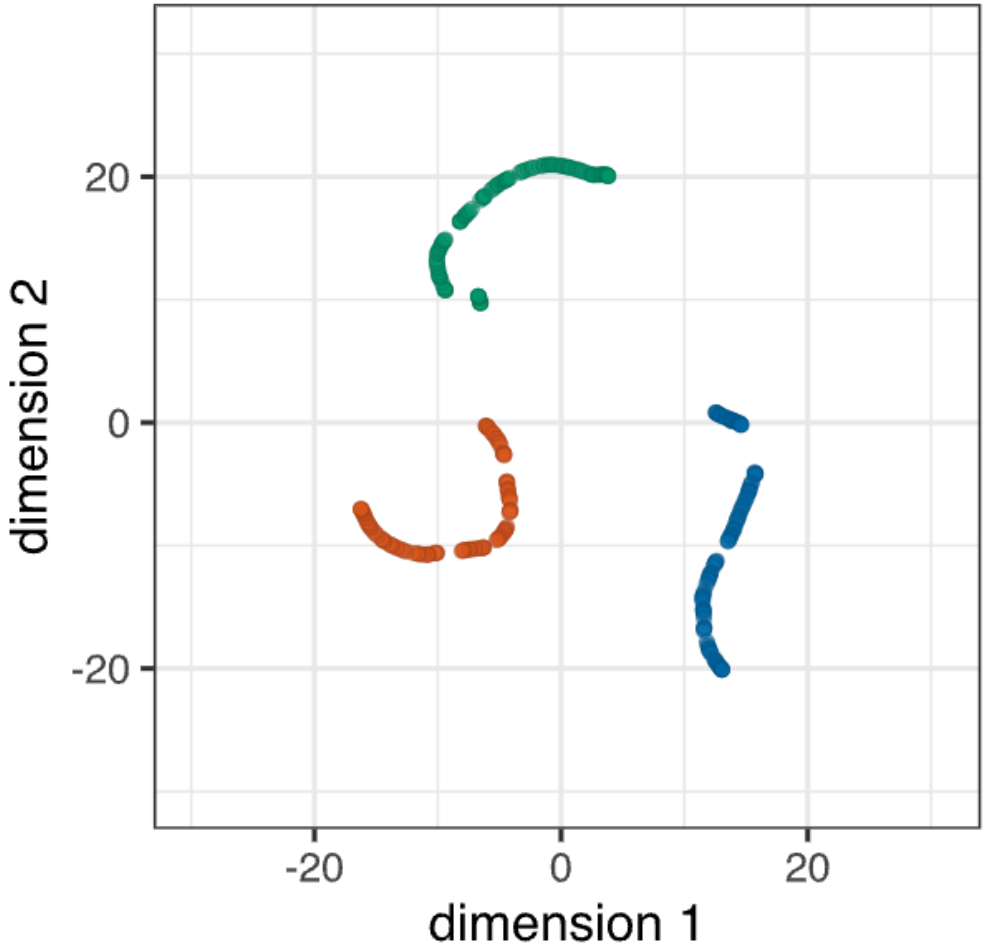
t-SNE results depend on the perplexity value



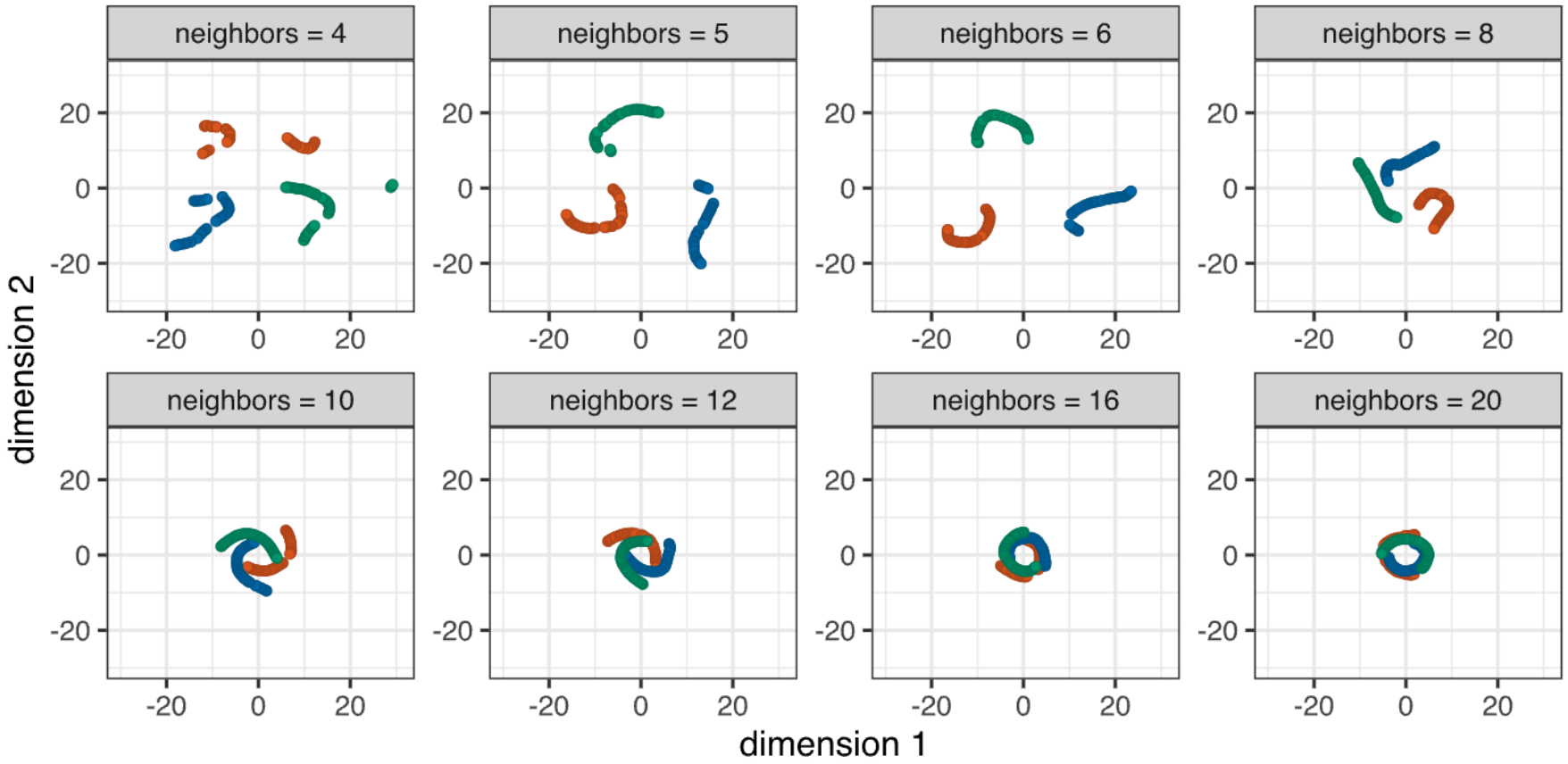
t-SNE results depend on the random starting point



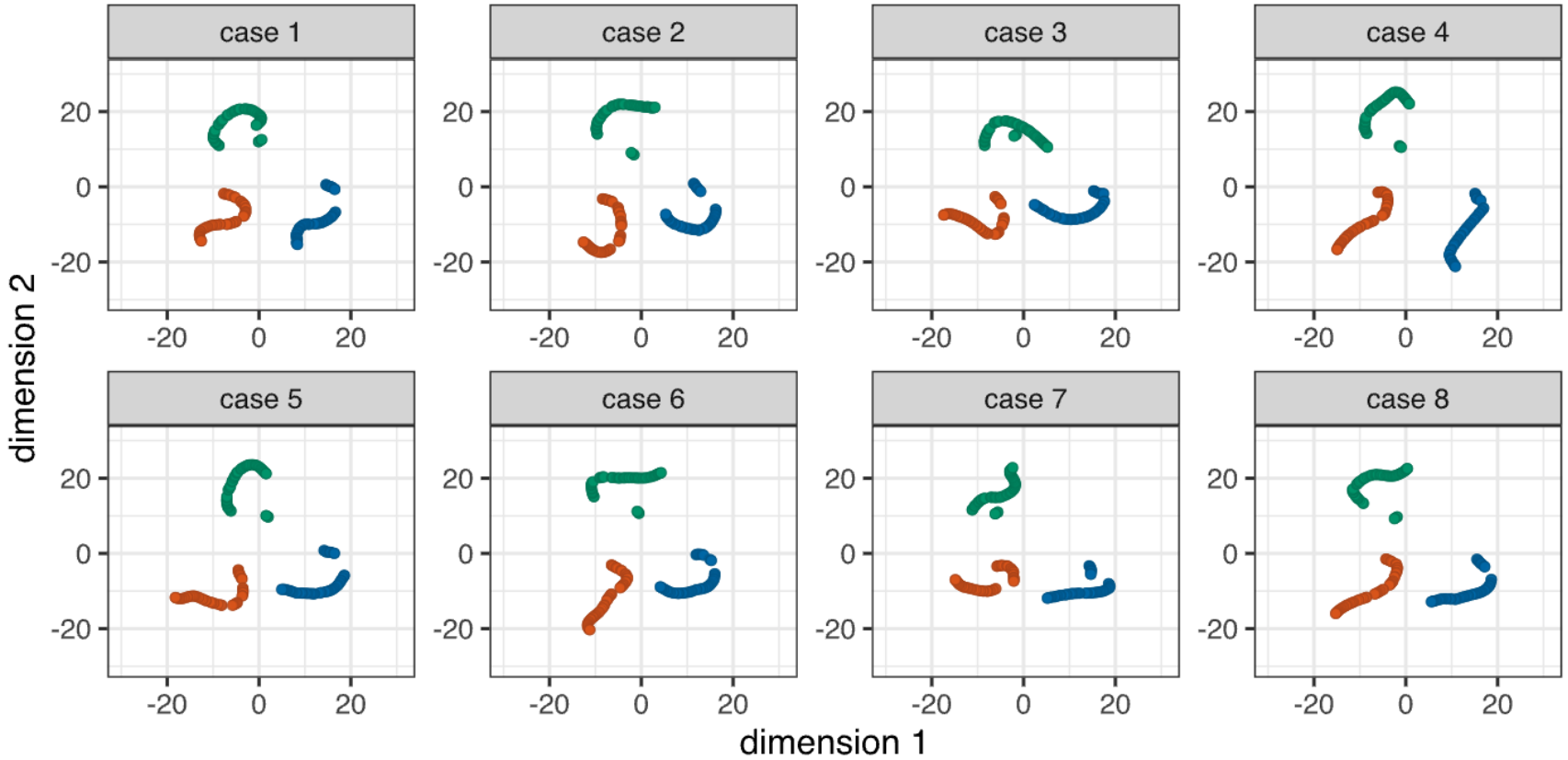
UMAP can separate the spirals too



UMAP results depend on the number of neighbors



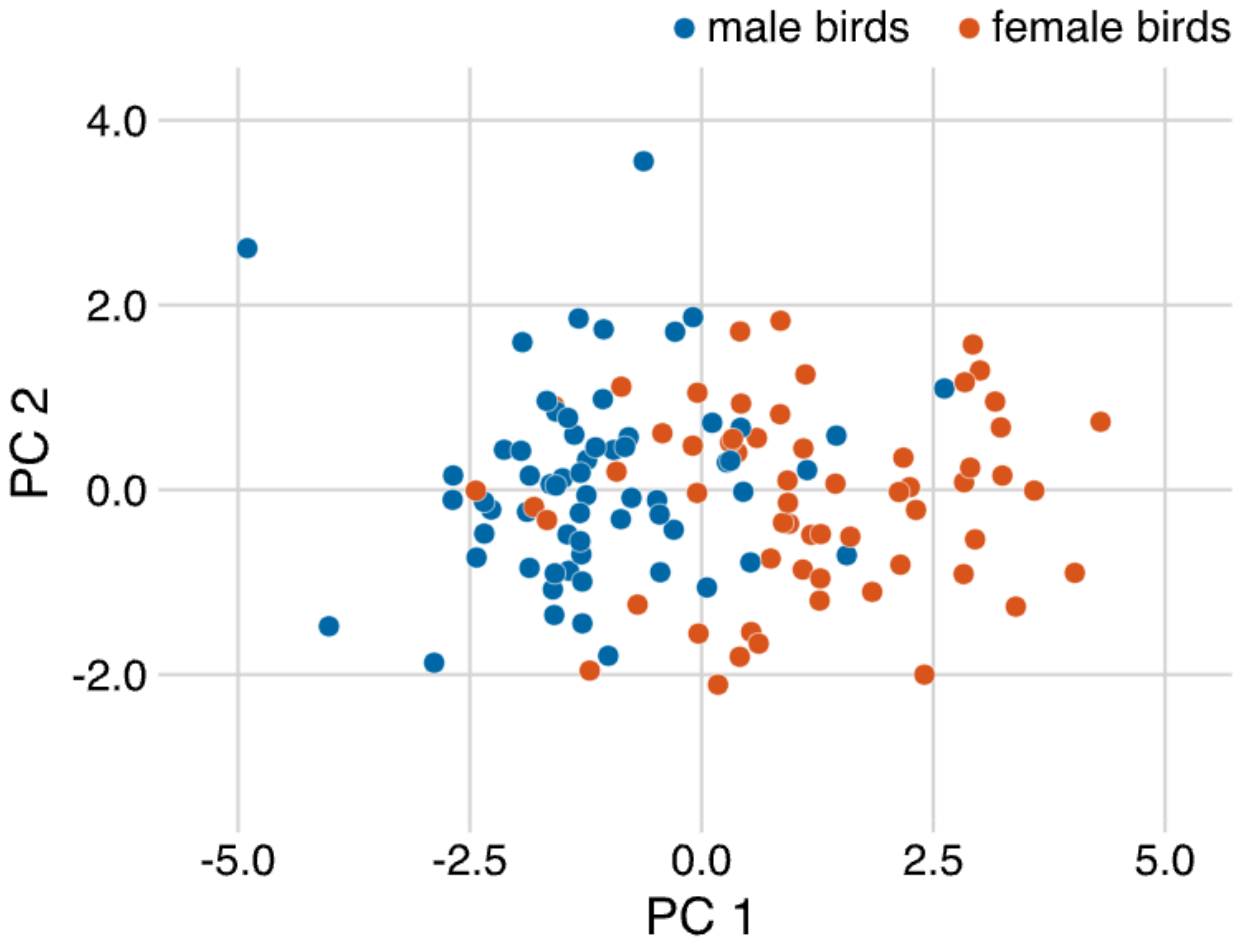
Random starting point has some impact on results



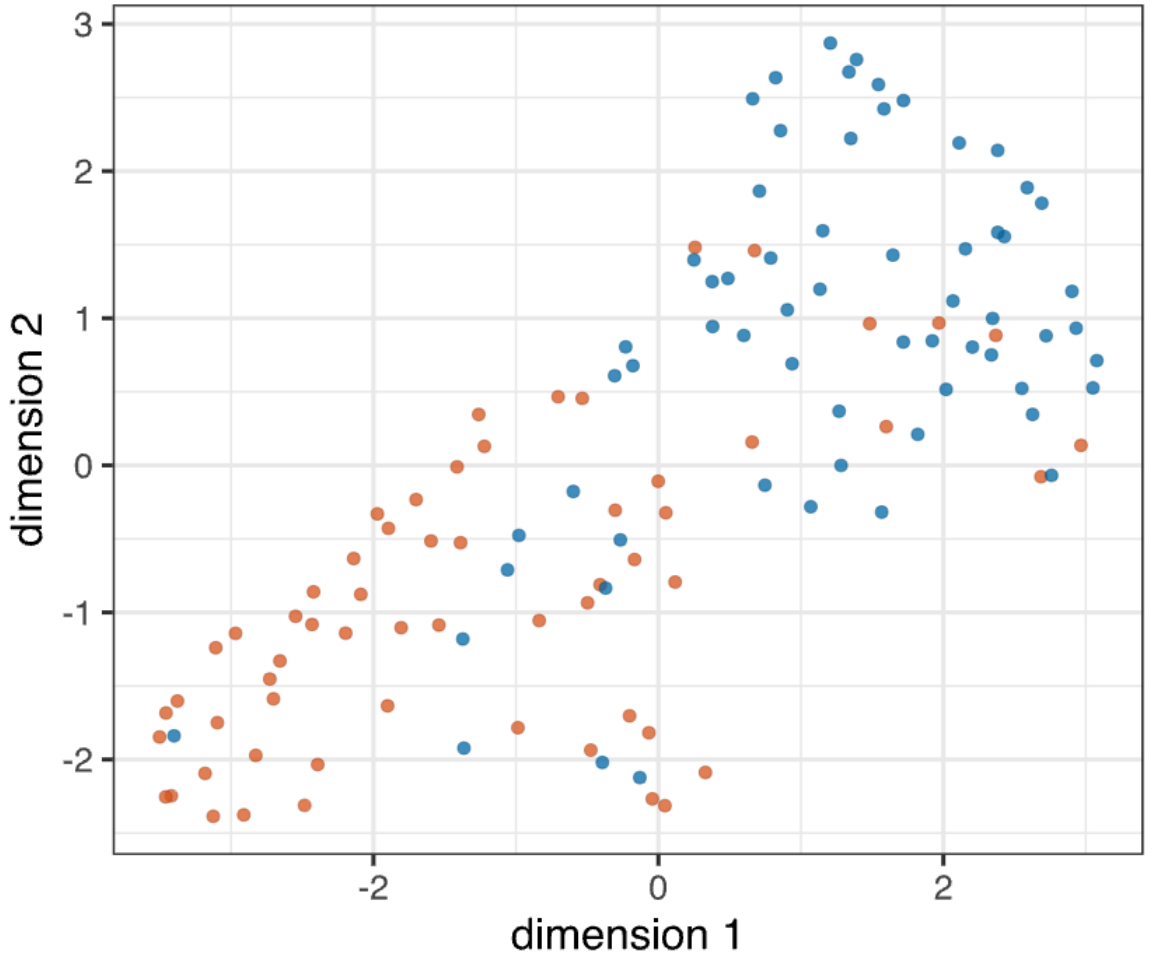
What is the meaning of the tuning parameters?

- Tuning parameters define when points are close in the original data space
- This implicitly defines the number of clusters generated
- These have comparable effects:
  - sigma (Gaussian kernel PCA)
  - perplexity (t-SNE)
  - number of neighbors (UMAP)

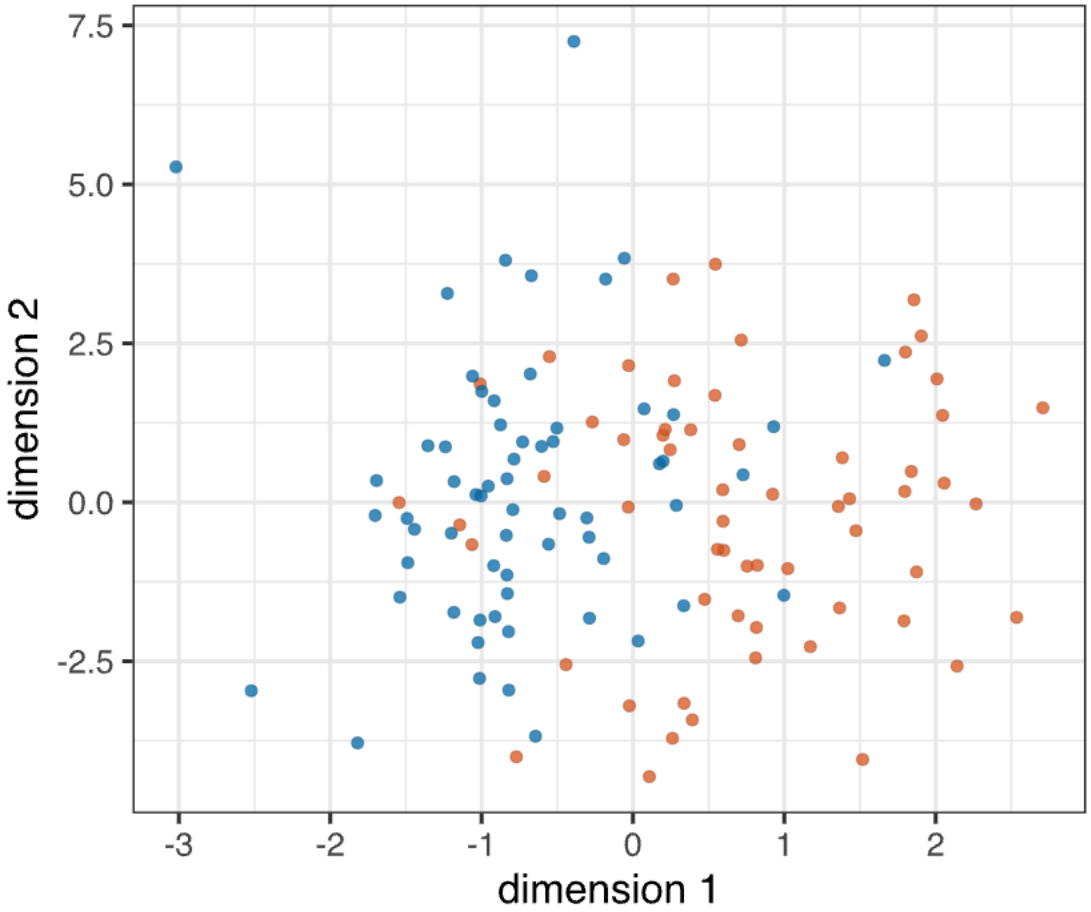
PCA on Blue Jay



UMAP on Blue Jay



Kernel PCA on Blue Jay



Nonlinear methods have important downsides

- Results depend on parameter fine tuning
- Low-dimensional embedding cannot be interpreted
  Use only when linear methods clearly aren't working