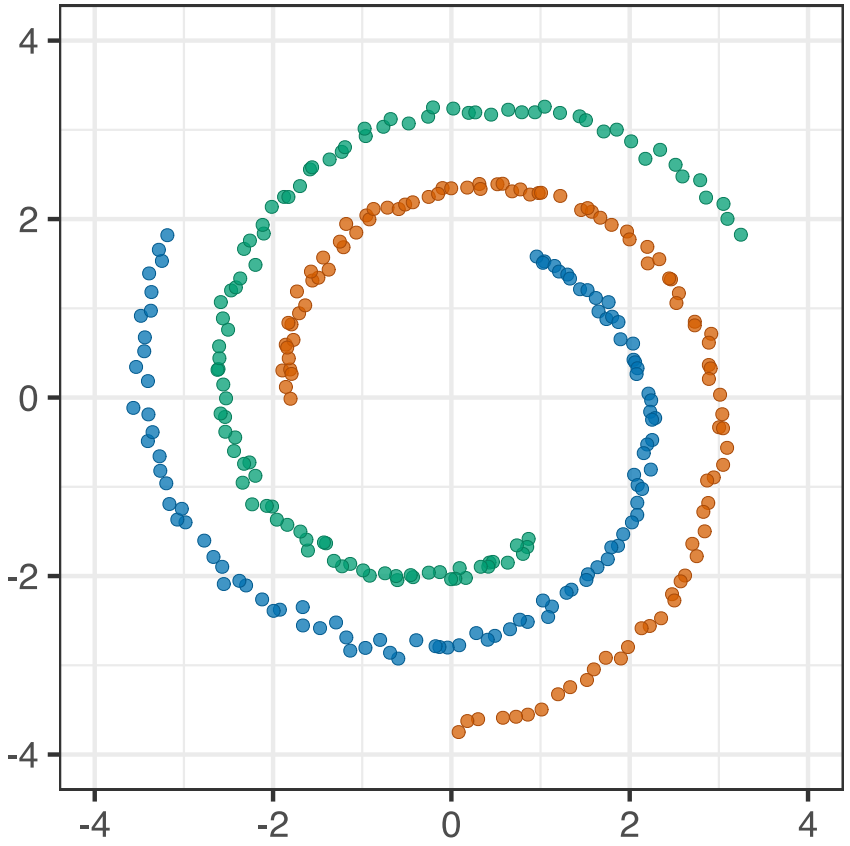
Dimension reduction 2

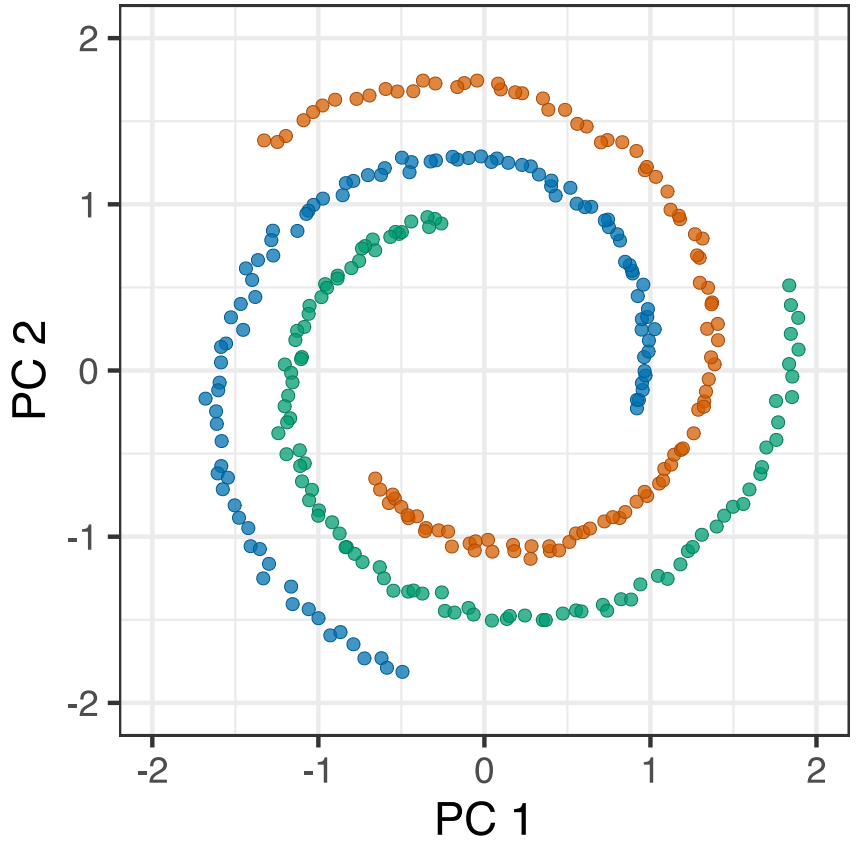
Claus O. Wilke

last updated: 2021-03-31

What if a rotation cannot disentangle the data?



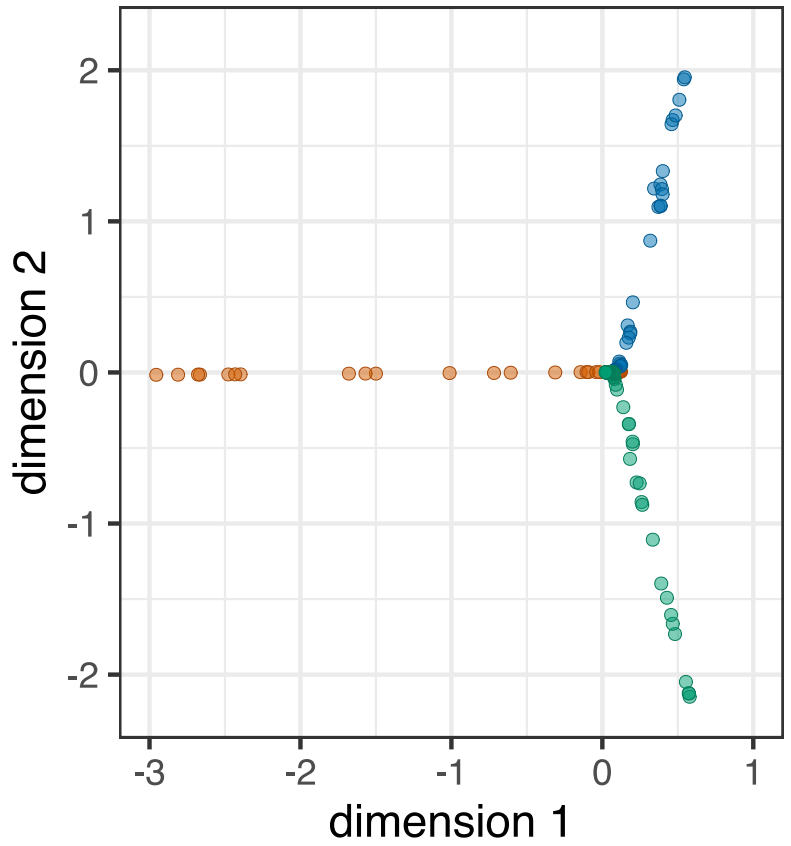
PCA analysis of intertwined spirals is not useful



One possible approach: Kernel PCA

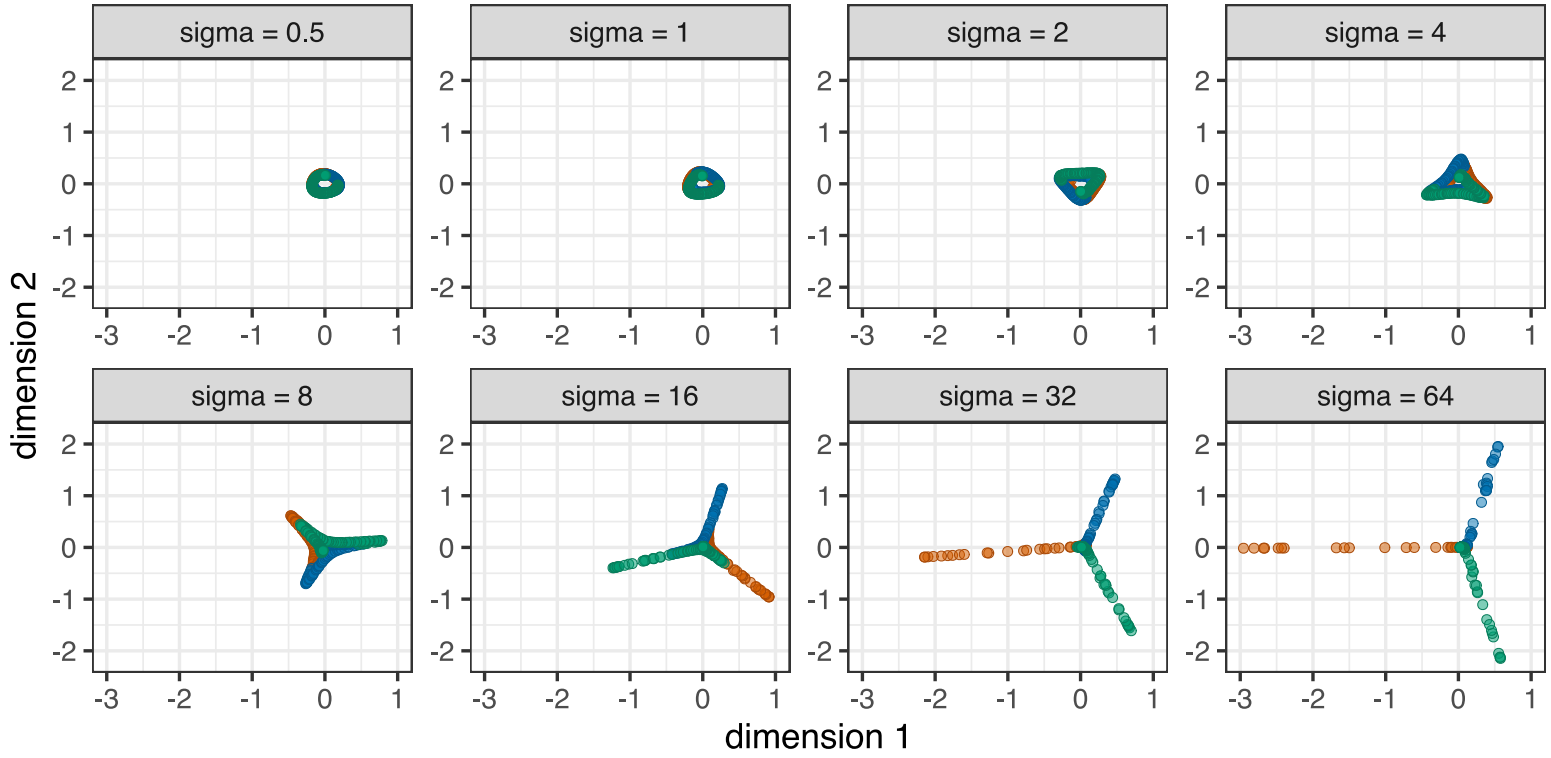
- Kernel PCA performs PCA in a hypothetical, higher-dimensional space
- With more dimensions, data points become more separable
- Importantly, the space is never explicitly constructed (kernel trick)
- Results from kernel PCA depend on choice of kernel

Kernel PCA can separate the spirals



Gaussian kernel, sigma = 64

But we need to choose the right sigma value

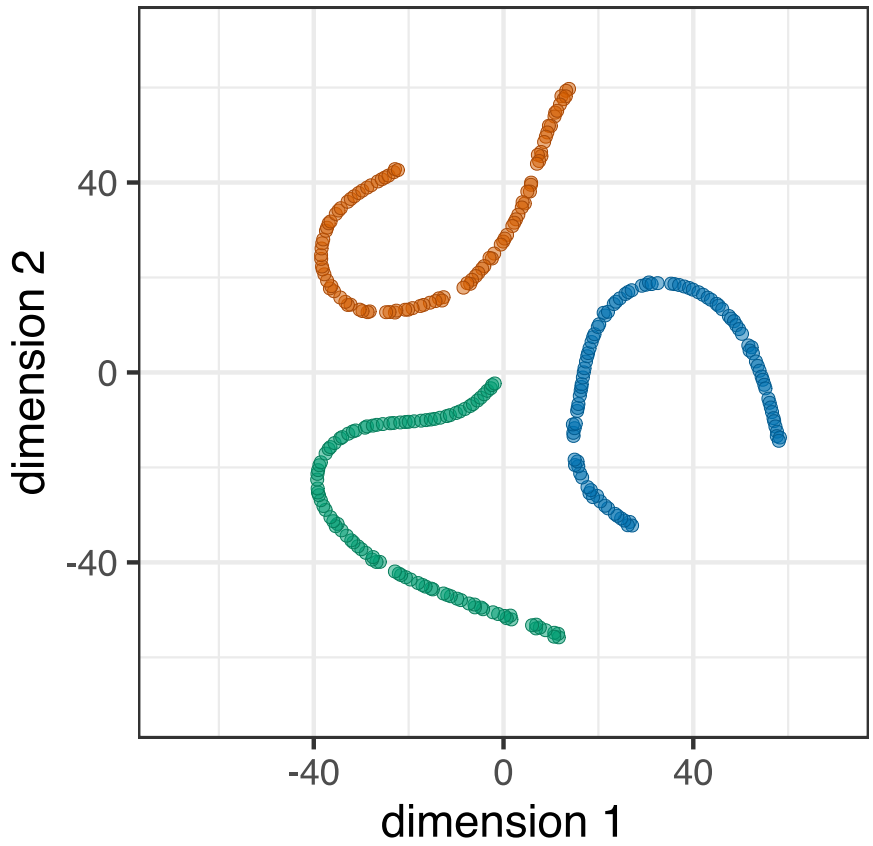


Other approaches

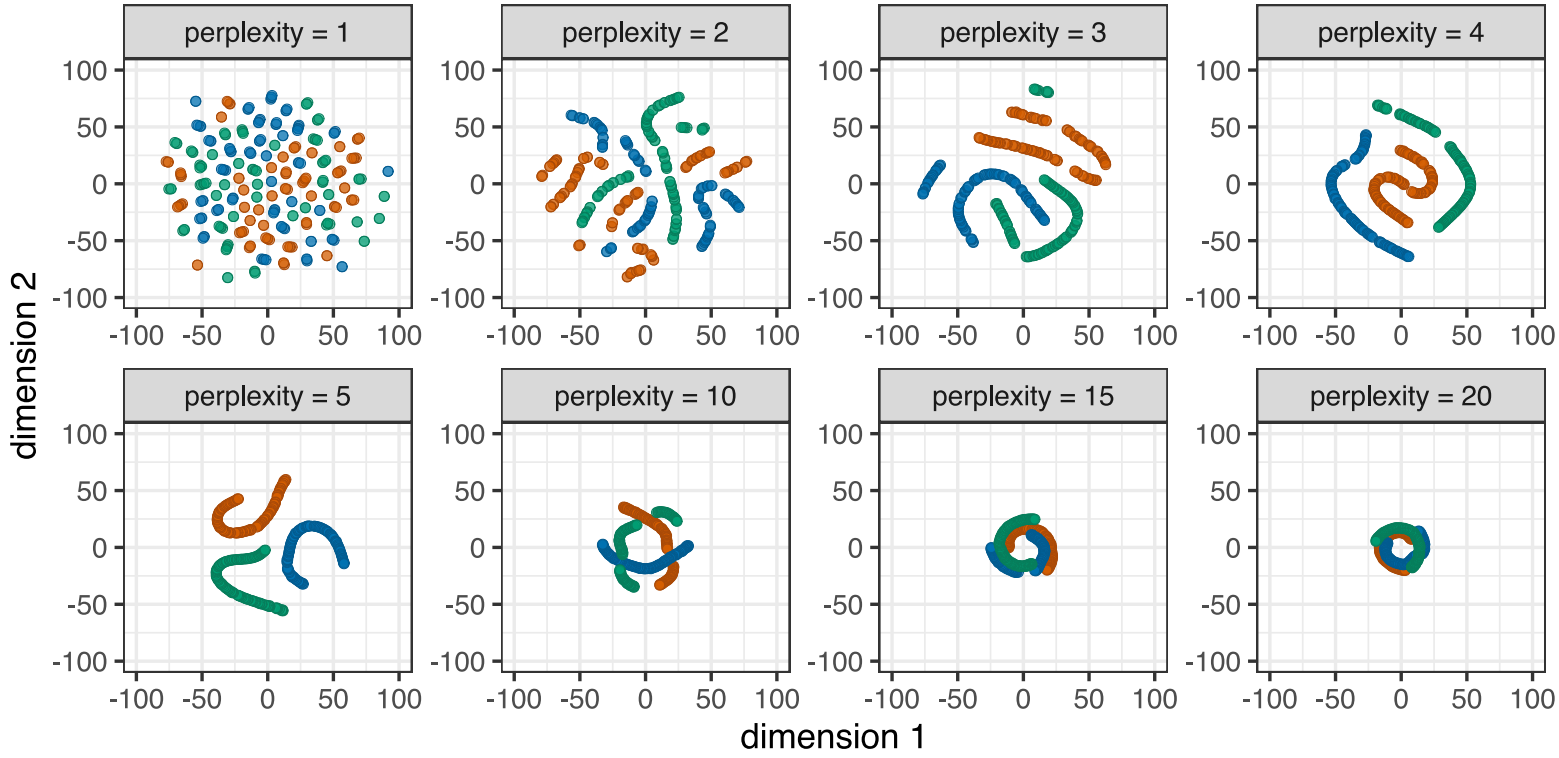
- t-SNE: t-distributed stochastic neighbor embedding
- UMAP: Uniform manifold approximation and projection

Both algorithms look at the local distances between points in the original data space and try to reproduce them in the low-dimensional representation

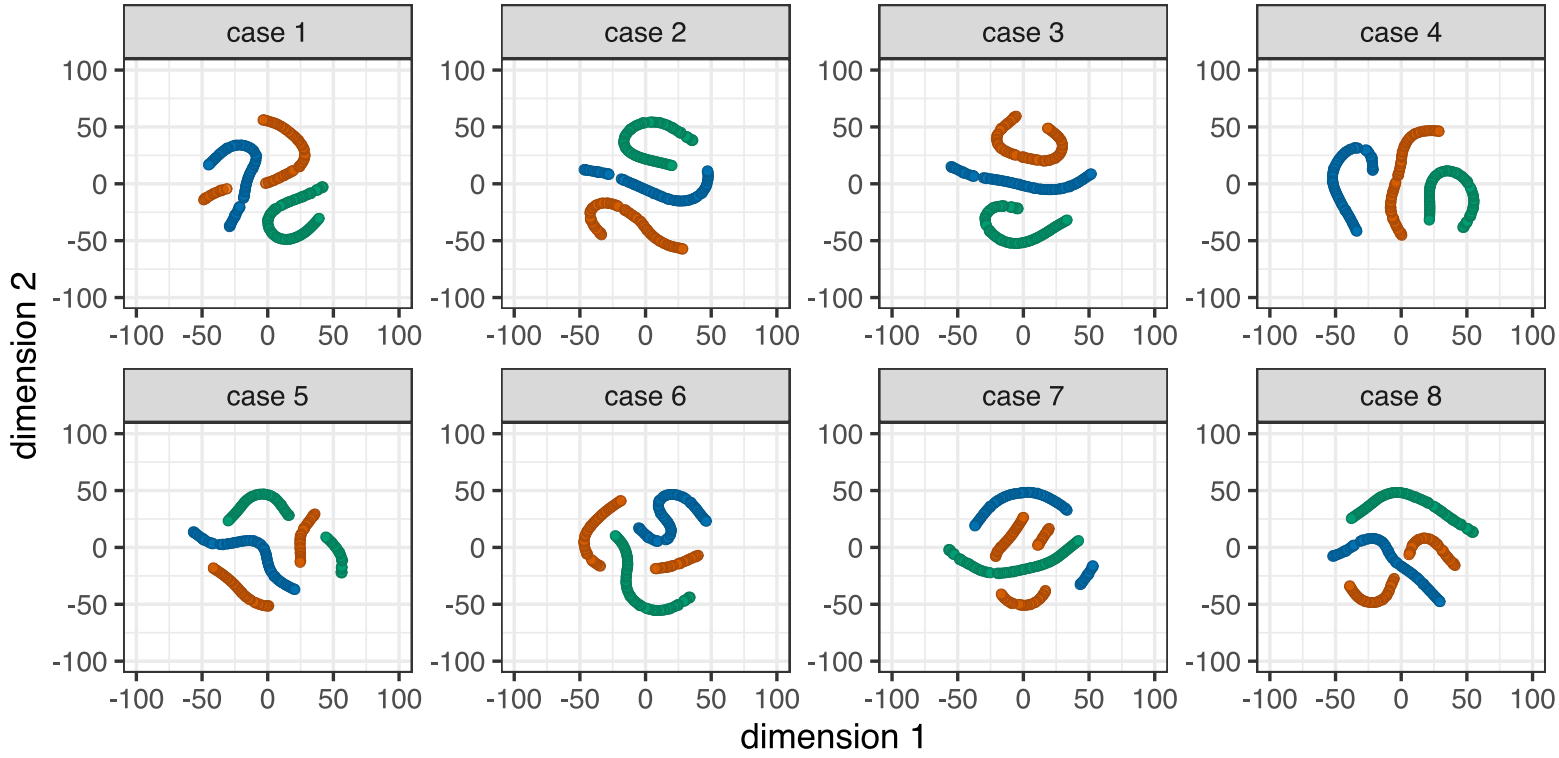
t-SNE can separate the spirals



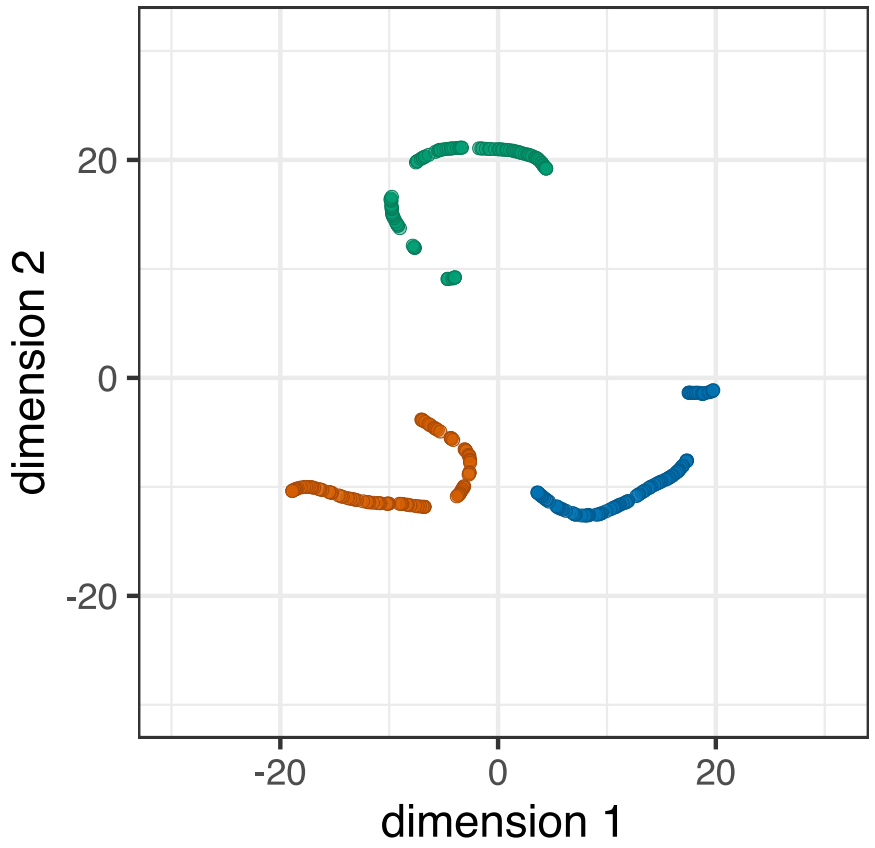
t-SNE results depend on the perplexity value



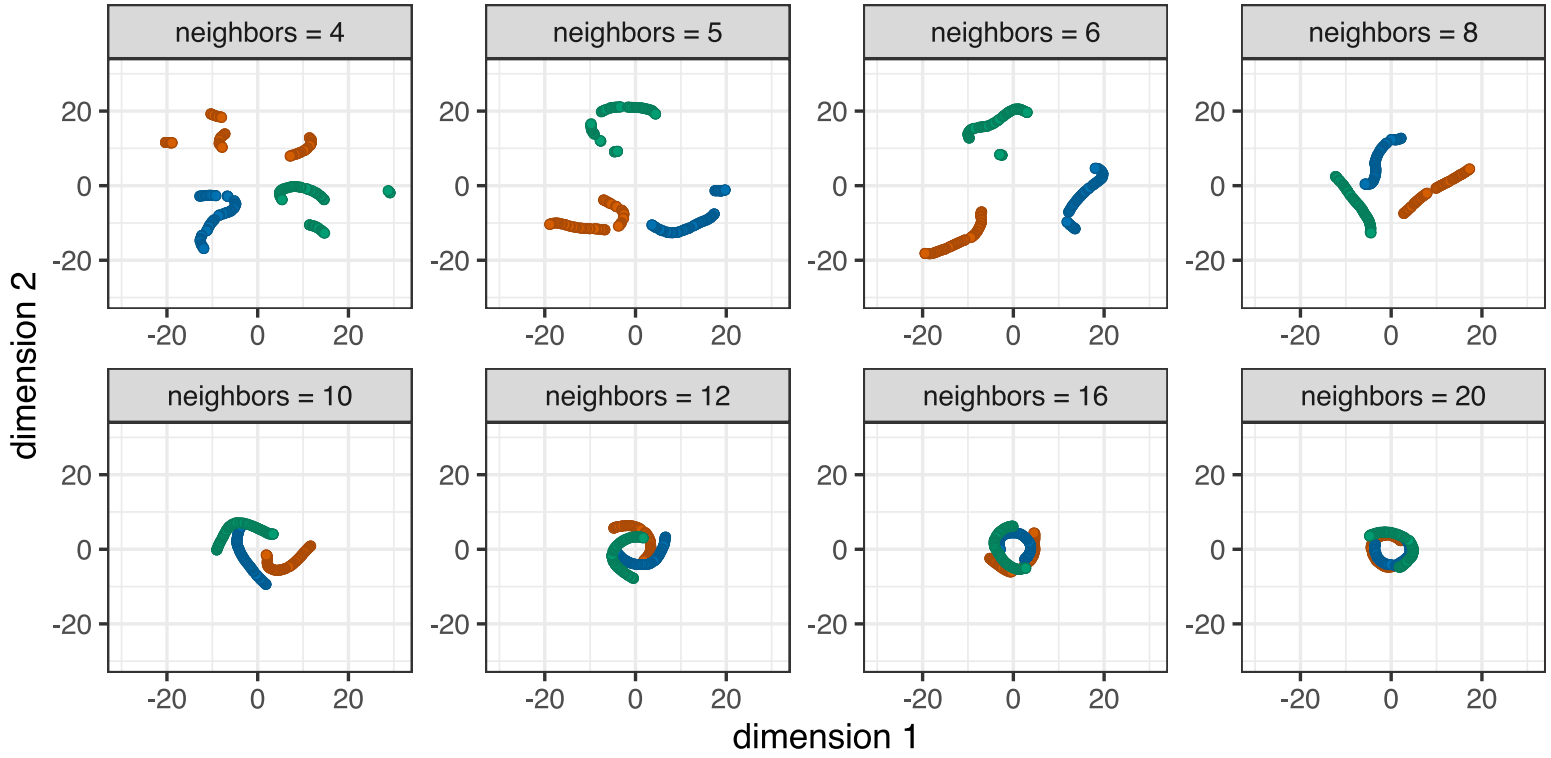
t-SNE results depend on the random starting point



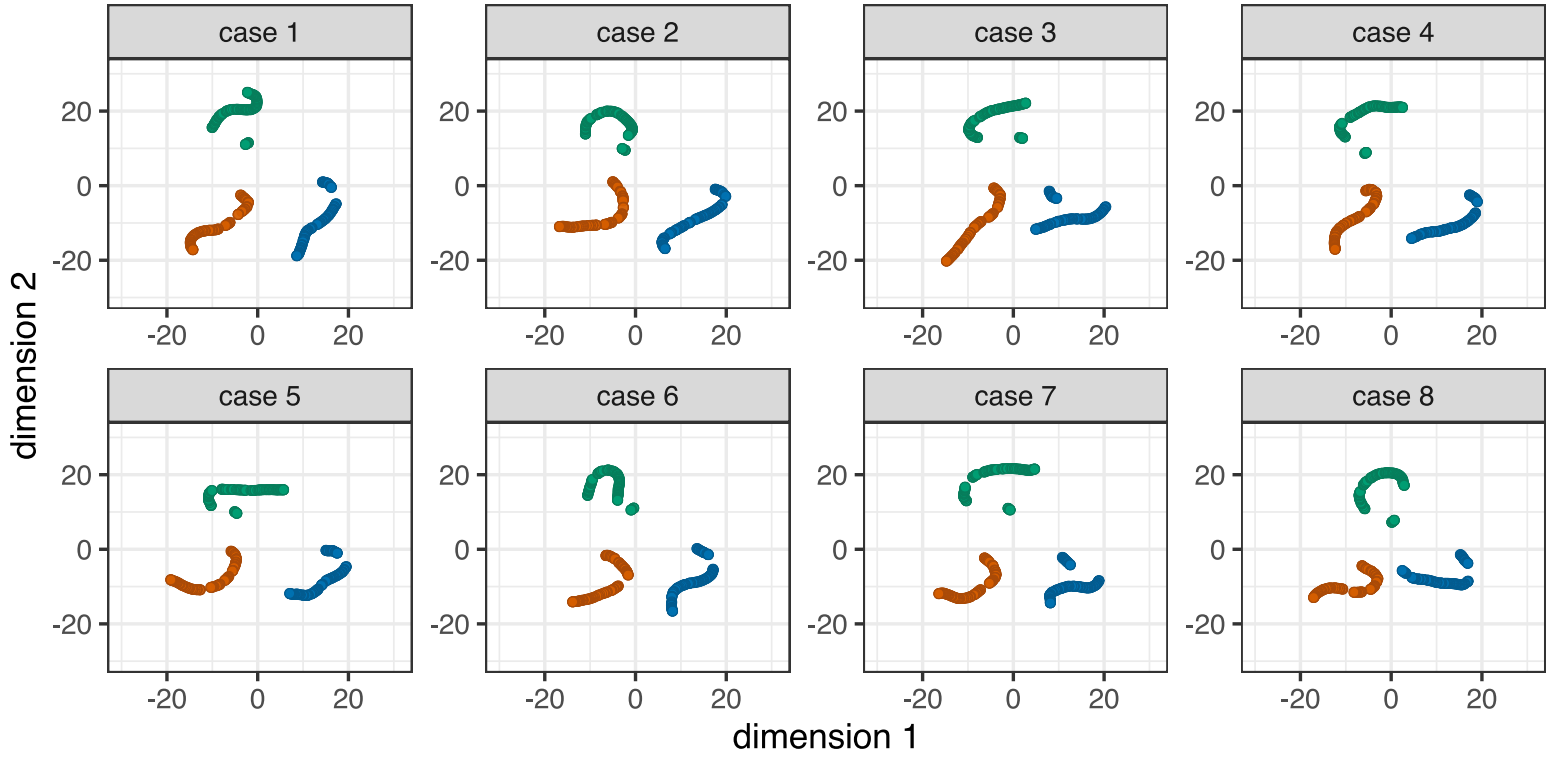
UMAP can separate the spirals



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

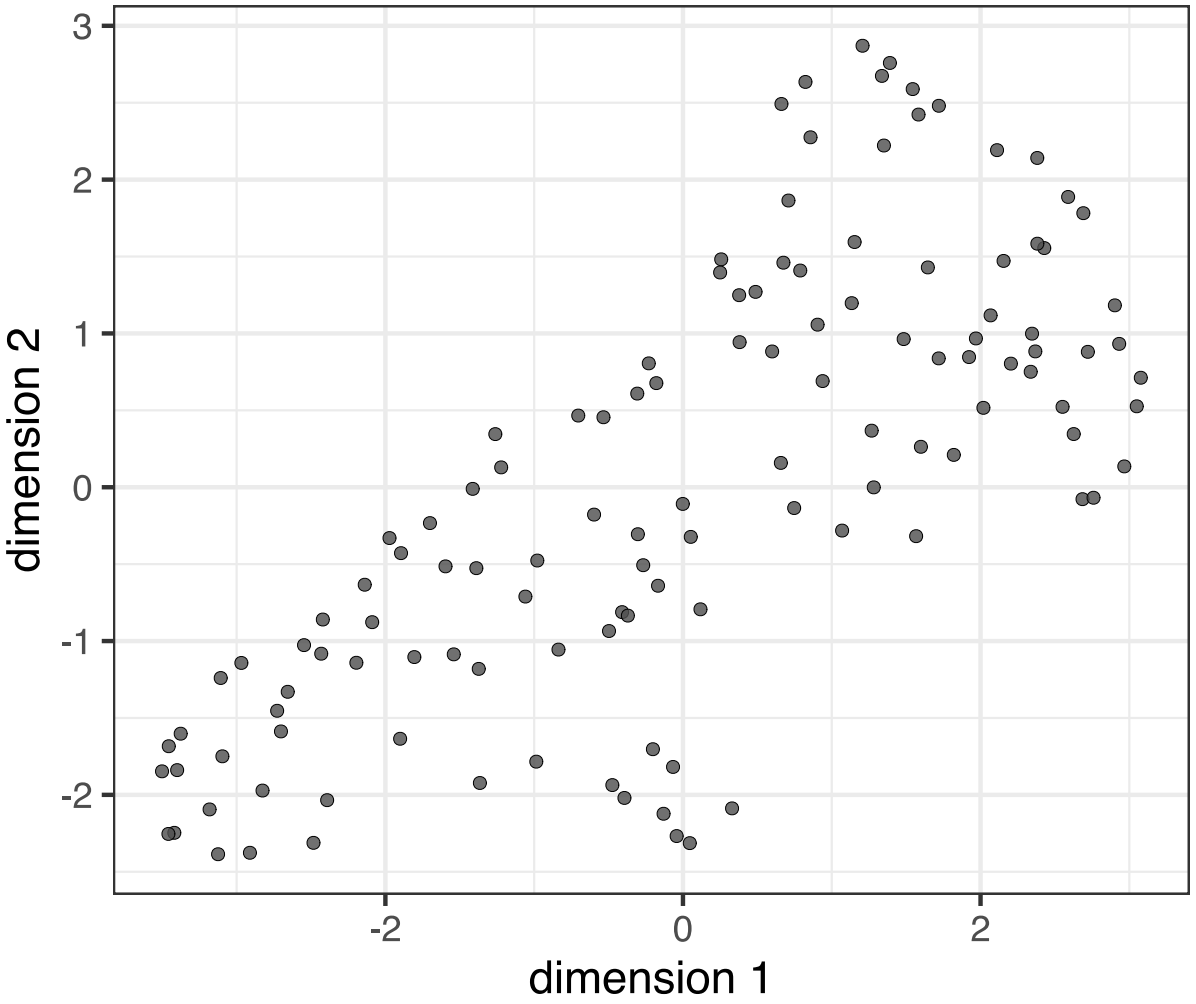
This implicitly defines the number of clusters generated

These have comparable effects:

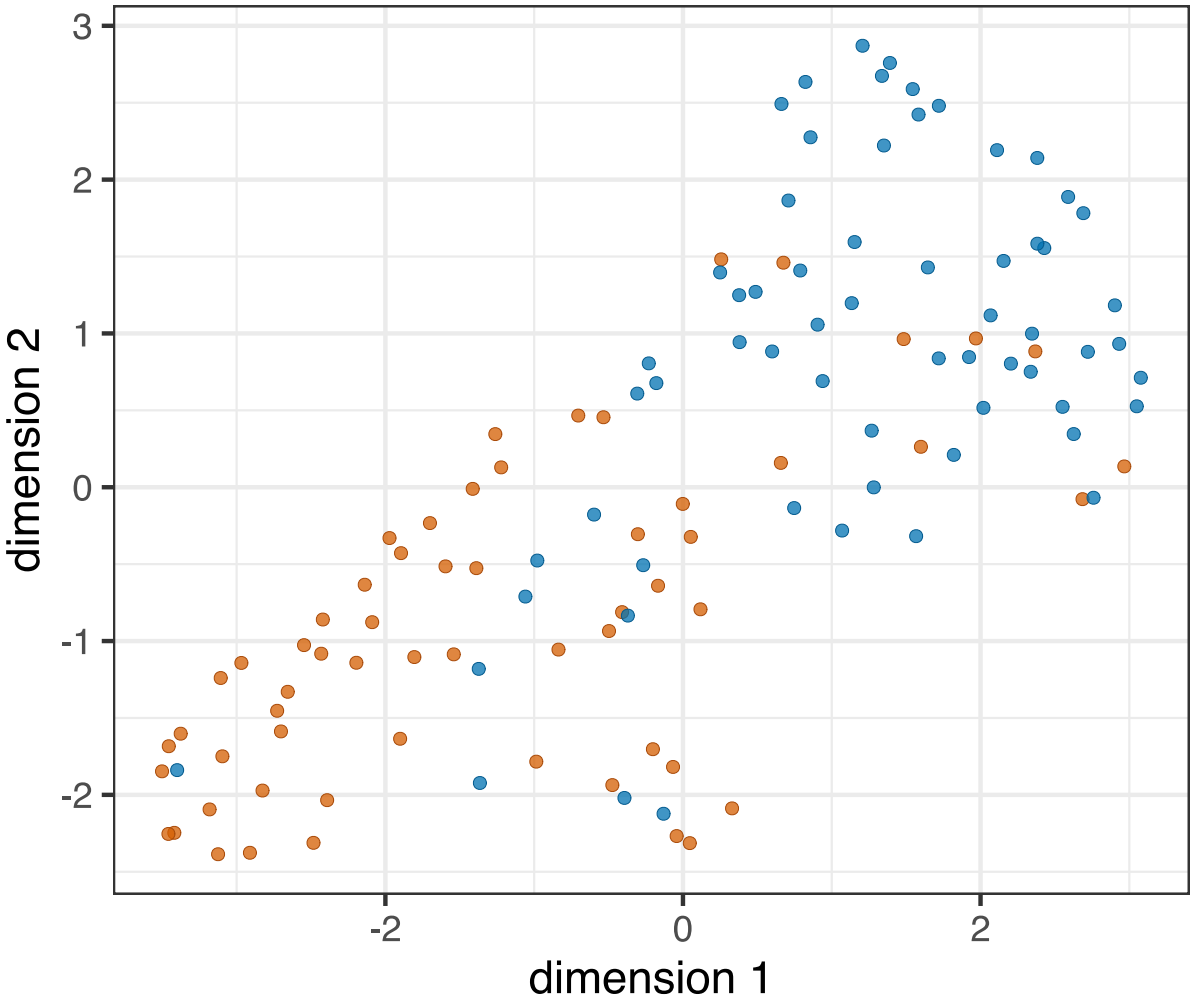
- sigma (Gaussian kernel PCA)
- perplexity (t-SNE)
- number of neighbors (UMAP)

How do these methods perform on the blue jays dataset?

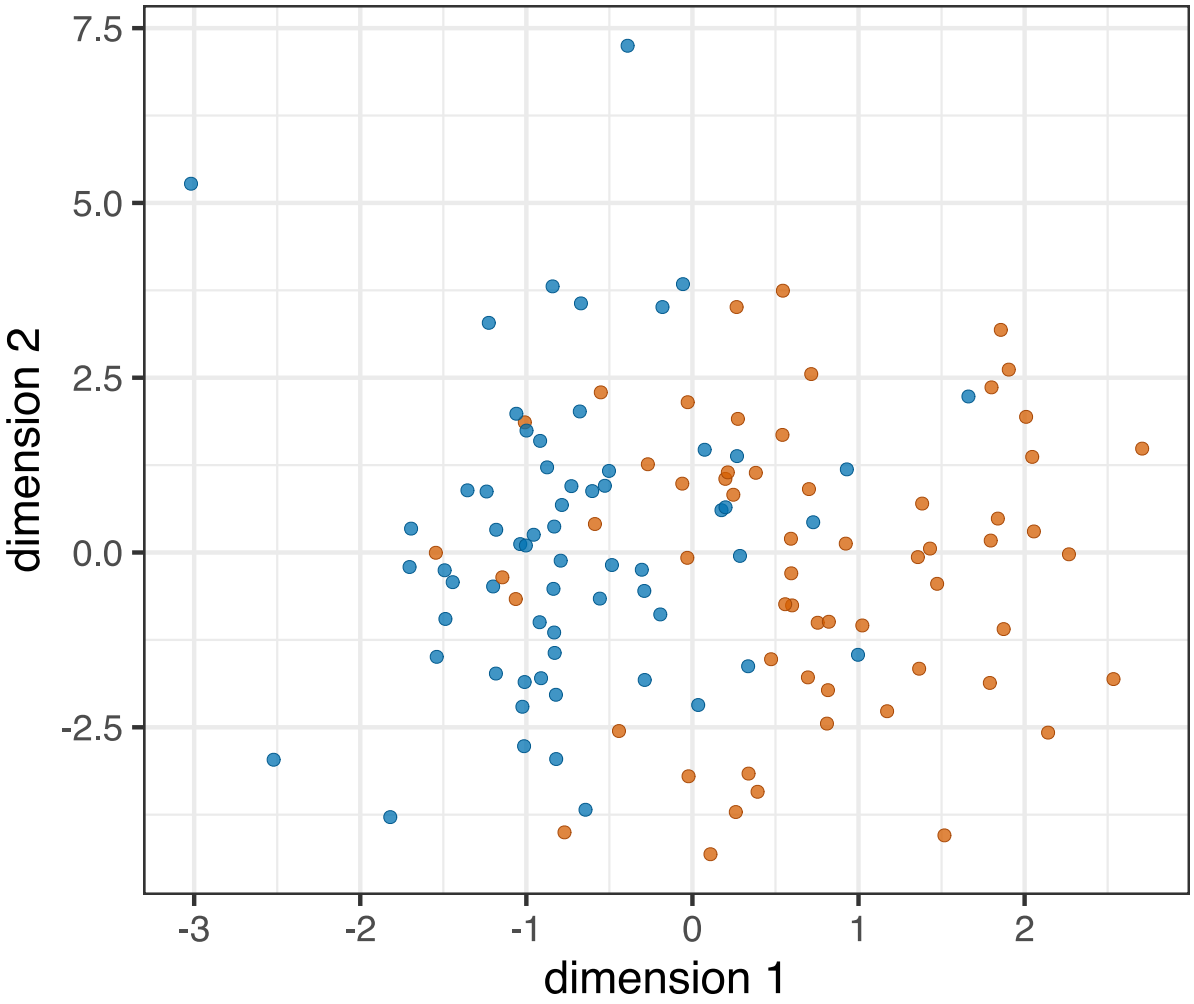
UMAP of blue jays



UMAP of blue jays



Kernel PCA of blue jays



Nonlinear methods have important downsides

- Results depend on parameter fine tuning
- Low-dimensional embedding cannot be interpreted (no rotation matrix plot)

Use only when linear methods clearly aren't working

Doing nonlinear dimension reduction in R

- All these methods require special packages:
  kernlab (kernel PCA)
  Rtsne (t-SNE)
  umap (UMAP)

- Code examples are somewhat messy

- Will do UMAP as example

Doing UMAP in R

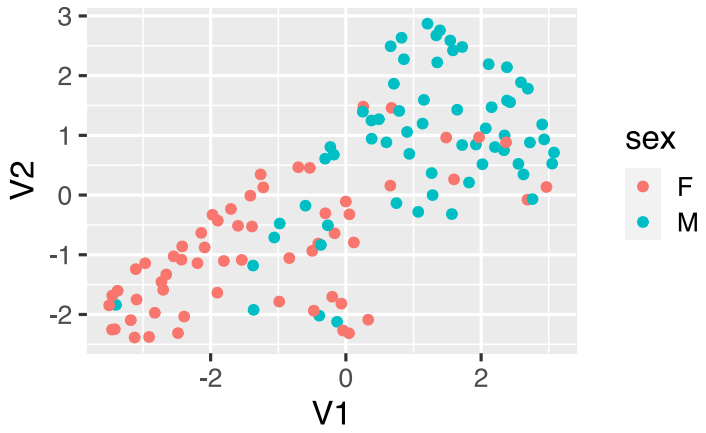
```
library(umap)

# set up UMAP parameters
custom.config <- umap.defaults
custom.config$n_neighbors <- 16         # number of neighbors
custom.config$n_epochs <- 500           # number of iterations for converg
custom.config$random_state <- 1234      # random seed

# calculate UMAP fit object
umap_fit <- blue_jays %>%
  select(where(is.numeric)) %>%         # retain only numeric columns
  scale() %>%                           # scale to zero mean and unit vari
  umap(config = custom.config)          # perform UMAP
```

Doing UMAP in R

```
# extract data and plot
umap_fit$layout %>%
  as.data.frame() %>%
  mutate(sex = blue_jays$sex) %>%
  ggplot(aes(V1, V2, color = sex)
  geom_point()
```



Further reading

- Wikipedia: Nonlinear dimensionality reduction
- Wikipedia: t-distributed stochastic neighbor embedding
- Wikipedia: Kernel principal component analysis
- **kernlab** reference documentation (for kernel PCA): pdf document
- **Rtsne** reference documentation: pdf document
- **umap** vignette: Uniform Manifold Approximation and Projection in R