TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

UMAP

Uniform Manifold Approximation and Projection

Anna Poetsch

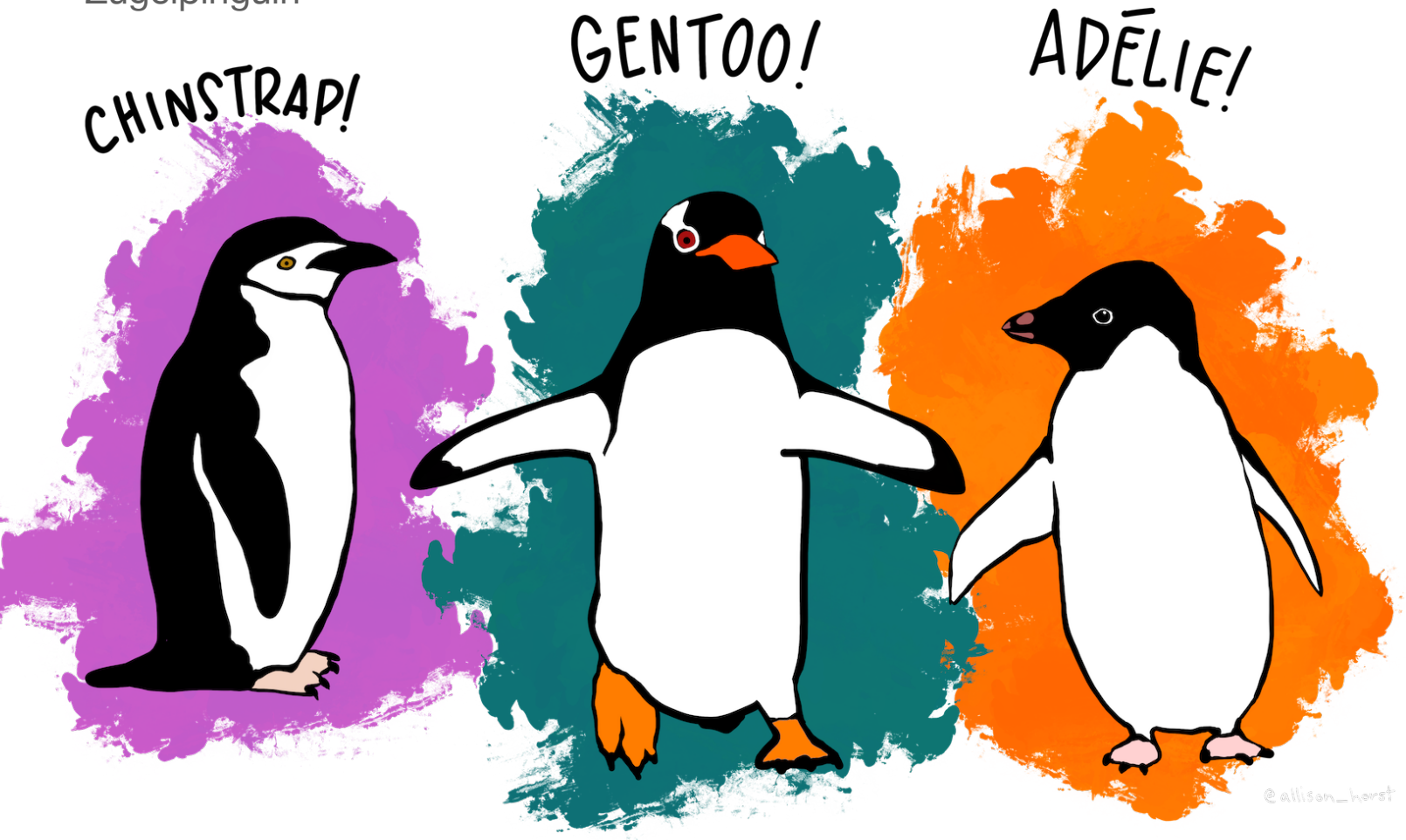
Research Group „Biomedical Genomics“, Biotechnology Center TU Dresden, NCT Dresden, and MSNZ

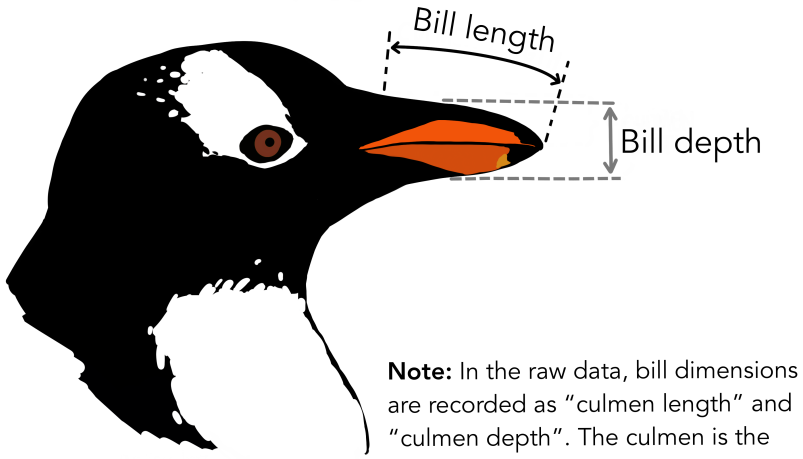
22.11.2021

@apoetsch

Zügelpinguin

Eselspinguin





Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

https://umap-learn.readthedocs.io/en/latest/

UMAP

Uniform Manifold Approximation and Projection for Dimension Reduction

Anna Poetsch
3. Nov 2021

Source material:
Tutorial: https://umap-learn.readthedocs.io/en/latest/
Paper: https://arxiv.org/abs/1802.03426
scRNA-Seq tutorial in Python: https://github.com/theislab/single-cell-tutorial
blood analysis in Python: https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html
blood analysis in R: https://satijalab.org/seurat/articles/pbmc3k_tutorial.html

Packages (if not available, pip install):

In [1]:
```
import numpy as np
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

Load data:

In [91]:
```
penguins = pd.read_csv("https://github.com/allisonhorst/palmerpenguins/raw/5b5891f01b52ae26ad8cb9755ec93672f49328a8/c
penguins = penguins.dropna()
```

Show data:

In [92]:
```
penguins.head()
```

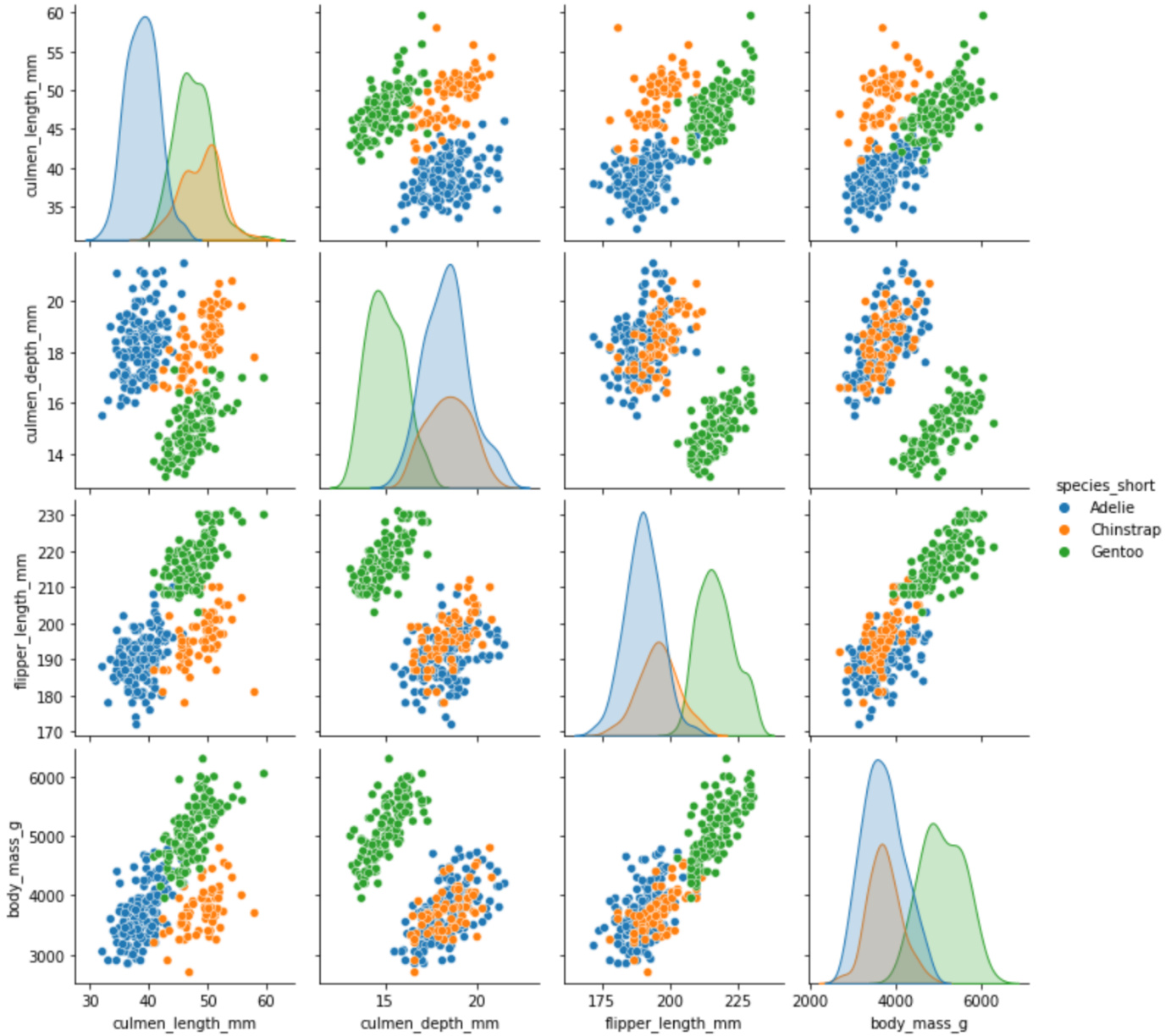
Out[92]:

	species_short	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	FEMALE
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	FEMALE
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	MALE

penguins = penguins.dropna() penguins.species_short.value_counts()

In [93]:
```
sns.pairplot(penguins, hue='species_short')
```

Out[93]: <seaborn.axisgrid.PairGrid at 0x146b49670>



change data format to values:

In [ ]:
```
penguin_data = penguins[
    [
        "culmen_length_mm",
        "culmen_depth_mm",
        "flipper_length_mm",
        "body_mass_g",
    ]
].values
```

UMAP

In [31]:
```
import umap.umap_ as umap #install with 'pip install umap-learn'
```

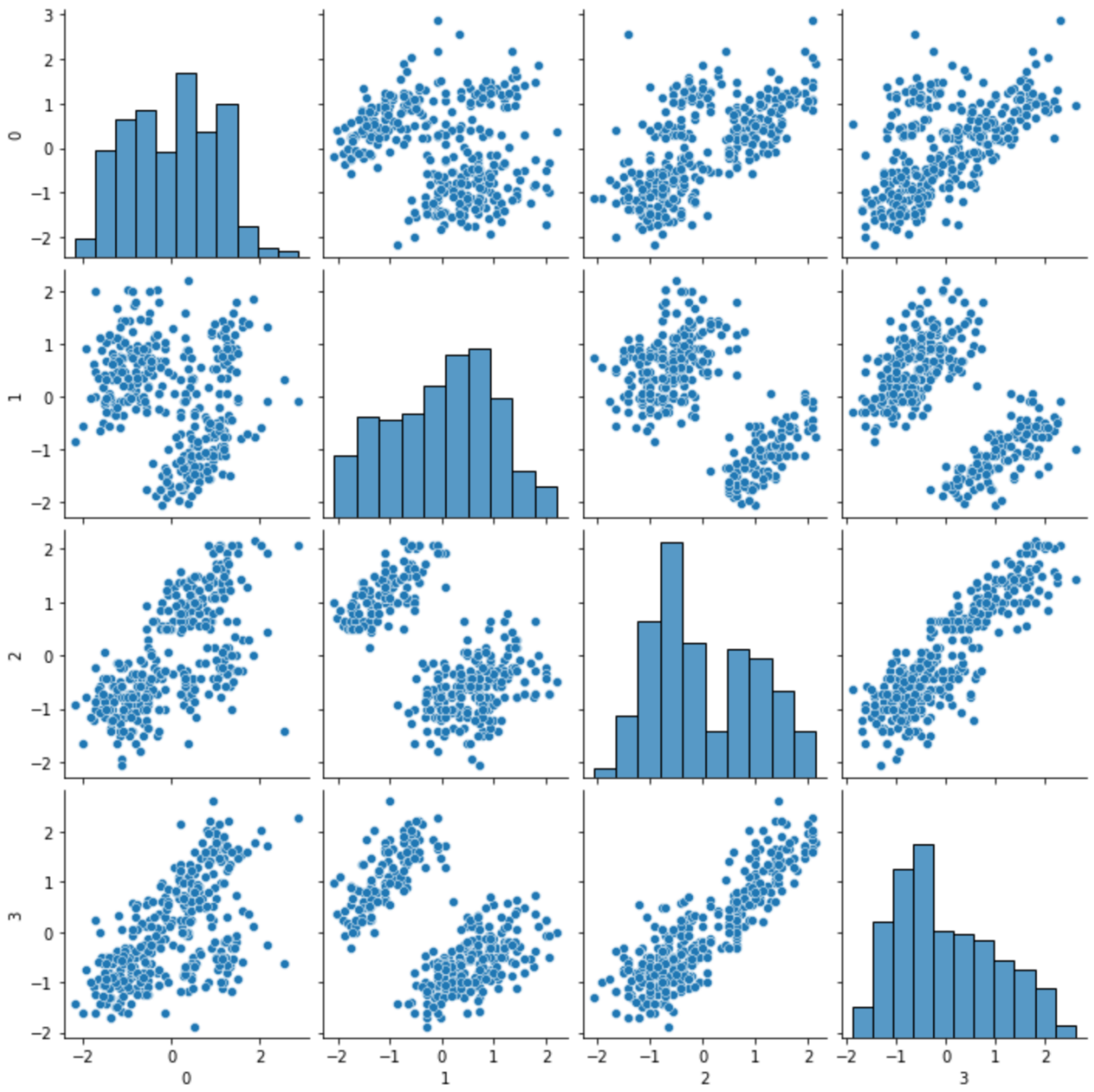
In [32]:
```
reducer = umap.UMAP()
```

In [9]:
```
scaled_penguin_data = StandardScaler().fit_transform(penguin_data)
```

In [98]:
```
scaled_df = pd.DataFrame(scaled_penguin_data)

sns.pairplot(scaled_df)
```

Out[98]: <seaborn.axisgrid.PairGrid at 0x1478f7f40>

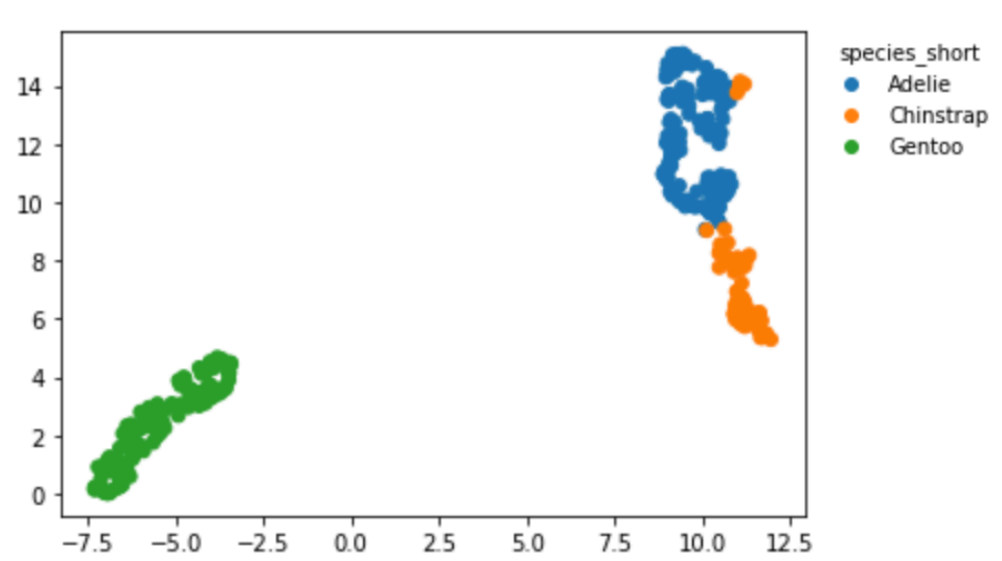


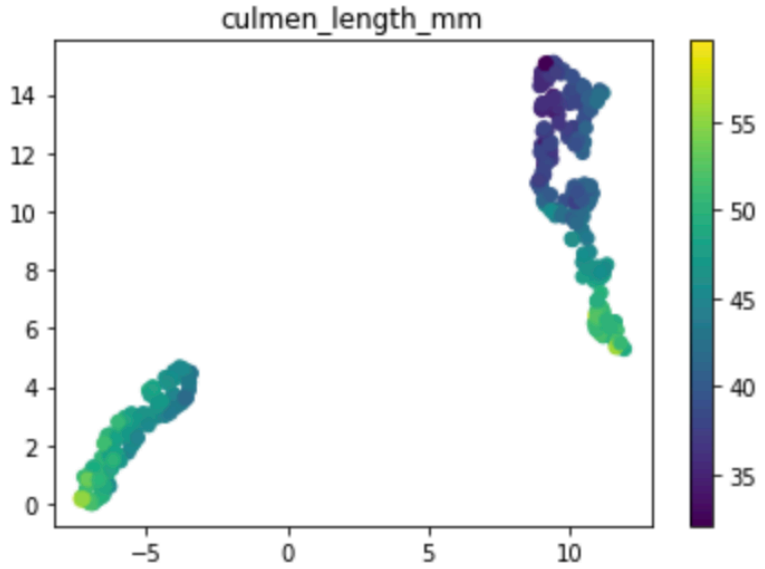
In [12]:
```
embedding = reducer.fit_transform(scaled_penguin_data)
embedding.shape
```

Out[12]:
```
(334, 2)
```

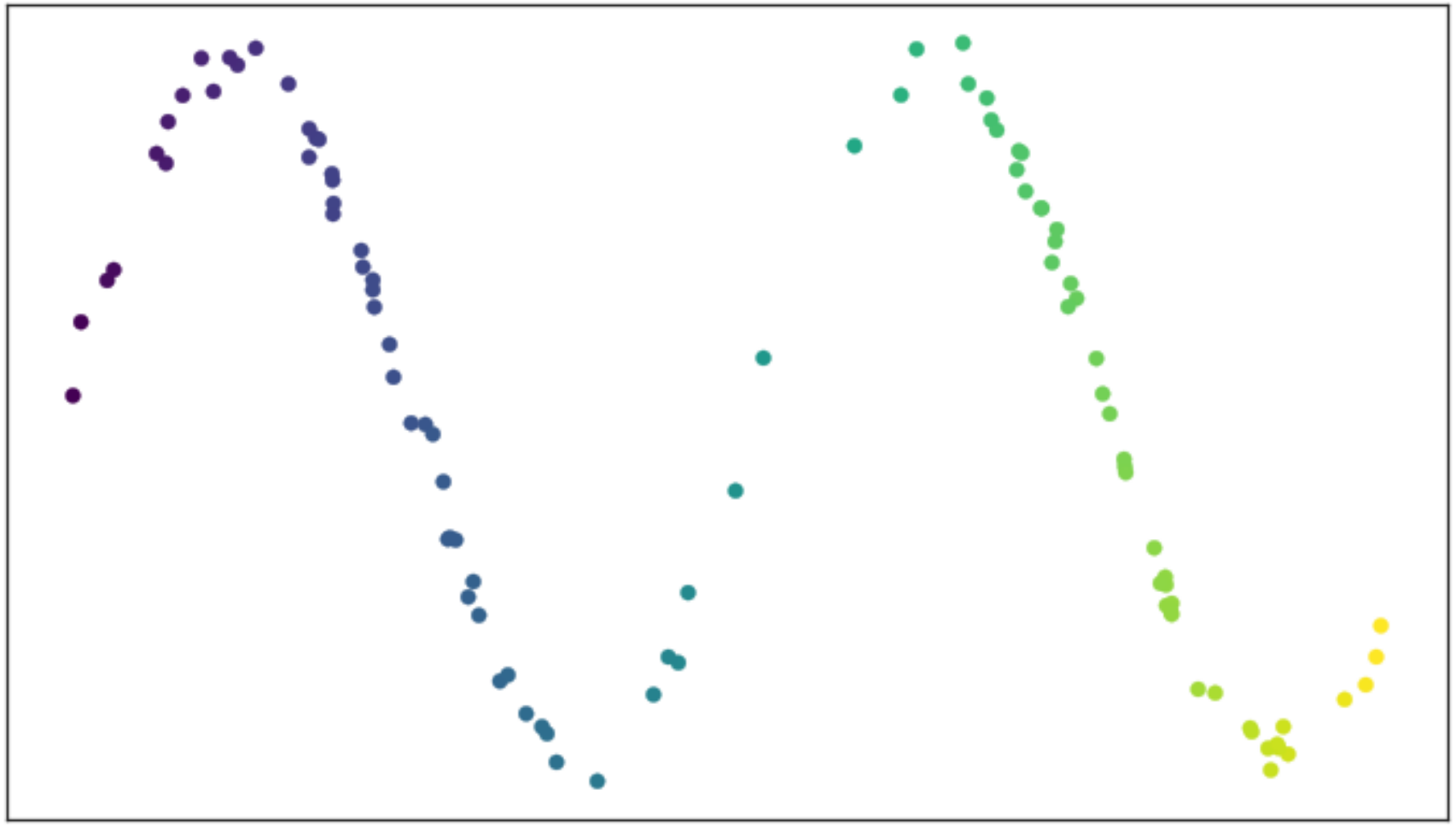
In [13]:
```
plt.scatter(
    embedding[:, 0],
    embedding[:, 1],
    c=[sns.color_palette()[x] for x in penguins.species_short.map({"Adelie":0, "Chinstrap":1, "Gentoo":2})])
```

Out[13]:
```
<matplotlib.collections.PathCollection at 0x14b3a9130>
```





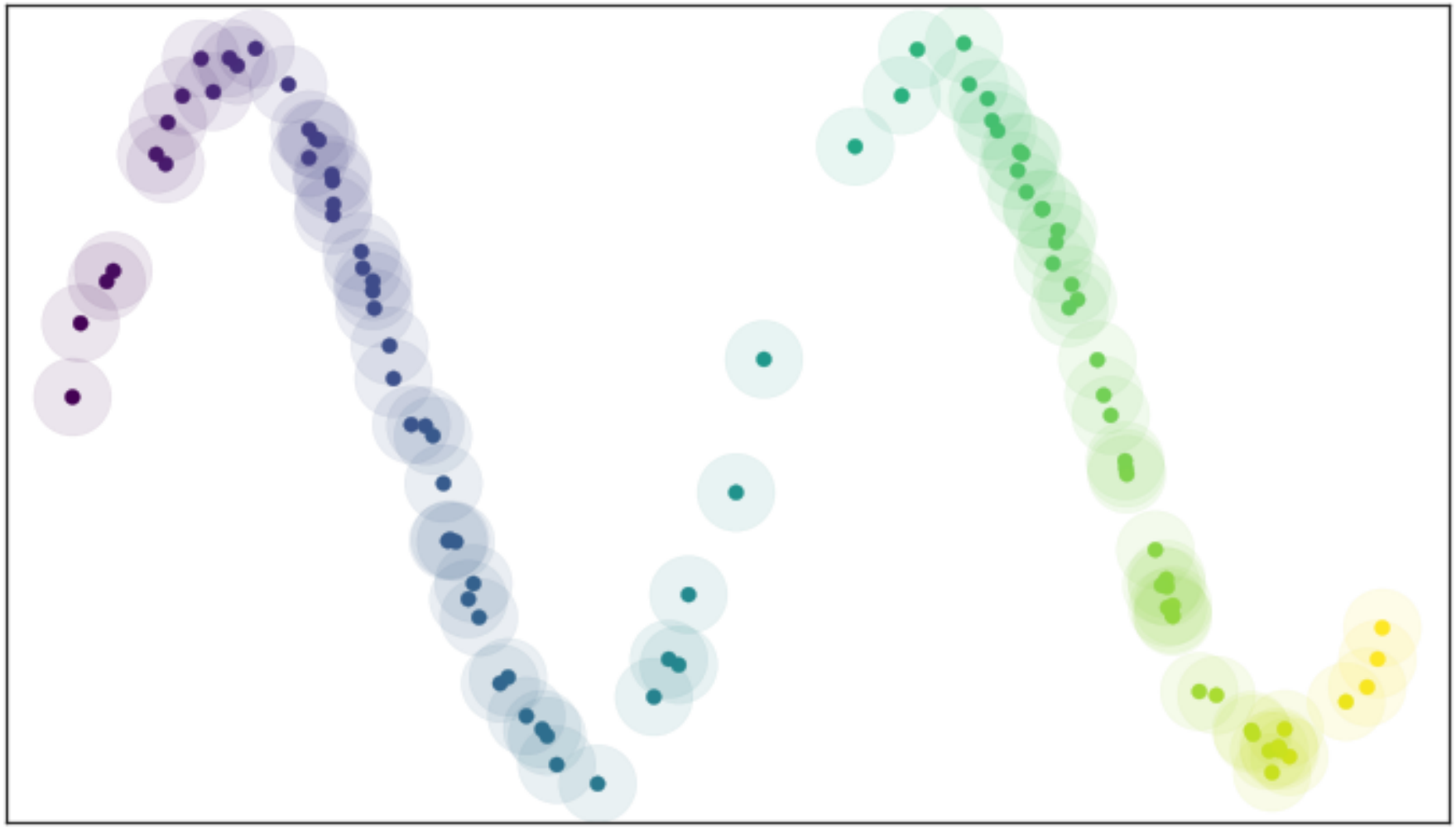
UMAP mit zwei Dimensionen



Als Beispiel eine Sinuskurve mit Rauschen

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

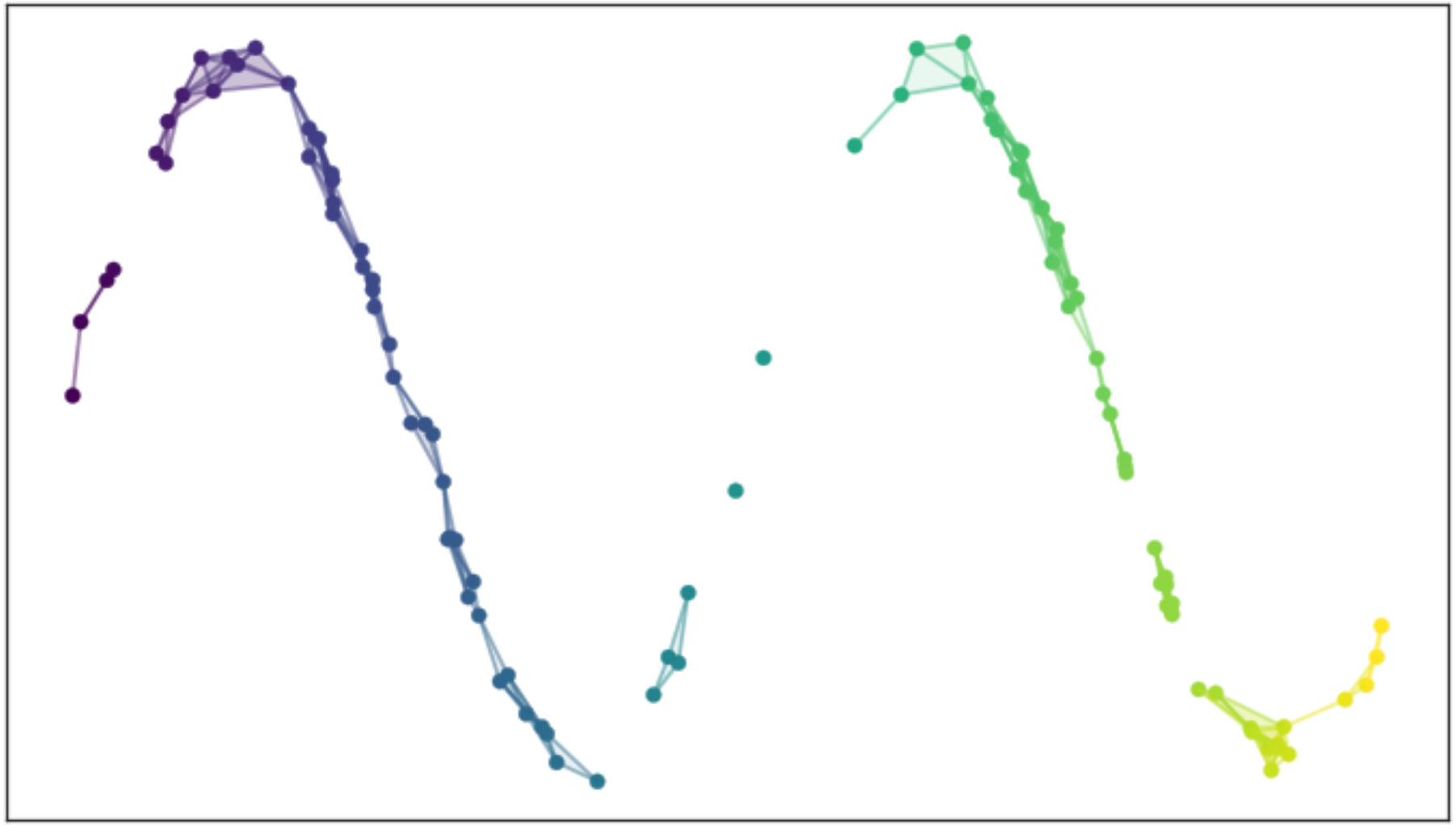
UMAP mit zwei Dimensionen



Ein Radius informiert uns ob Nachbarn da sind.

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

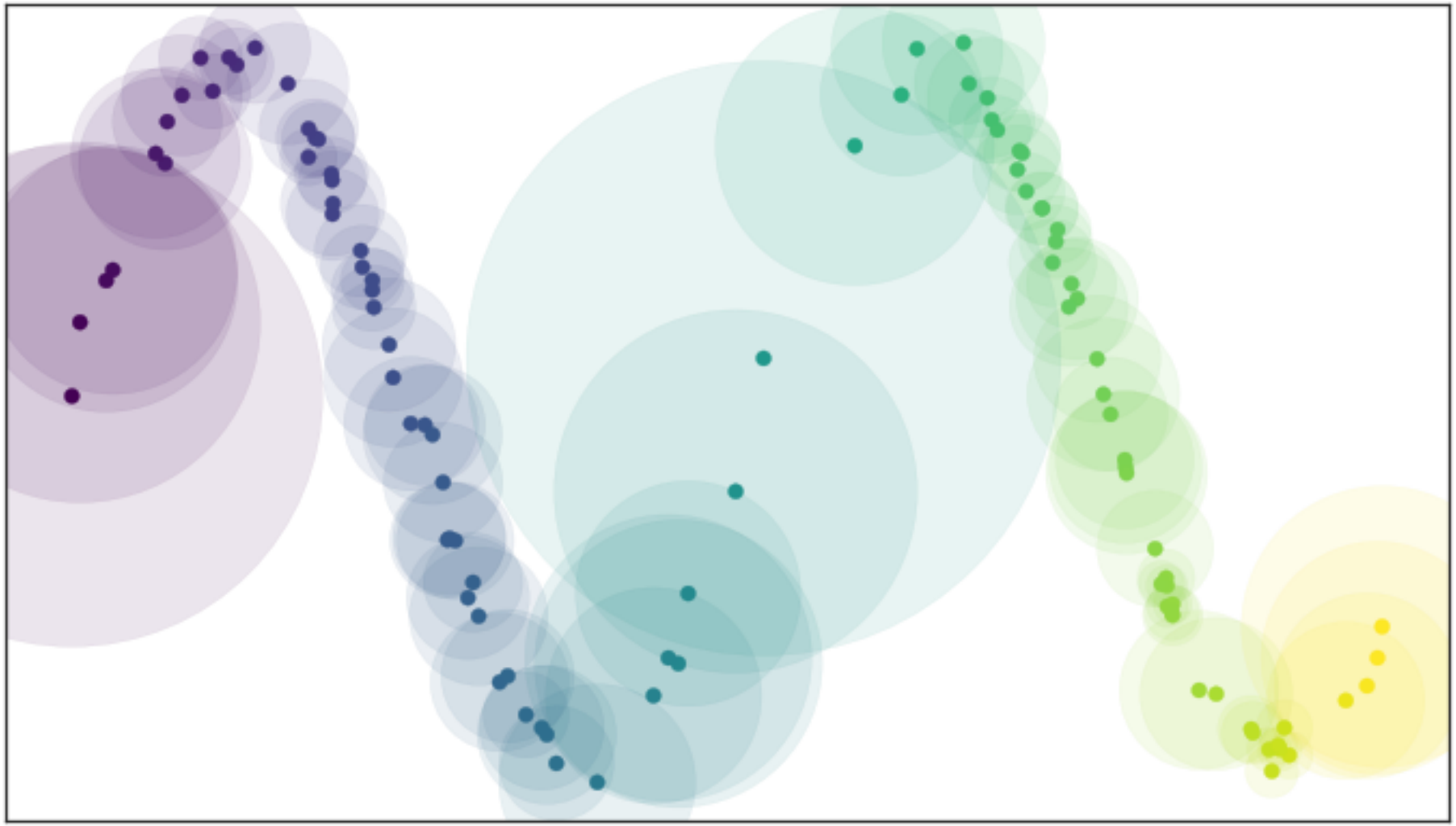
UMAP mit zwei Dimensionen



Die resultierende ja-nein-Antwort ist etwas suboptimal

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

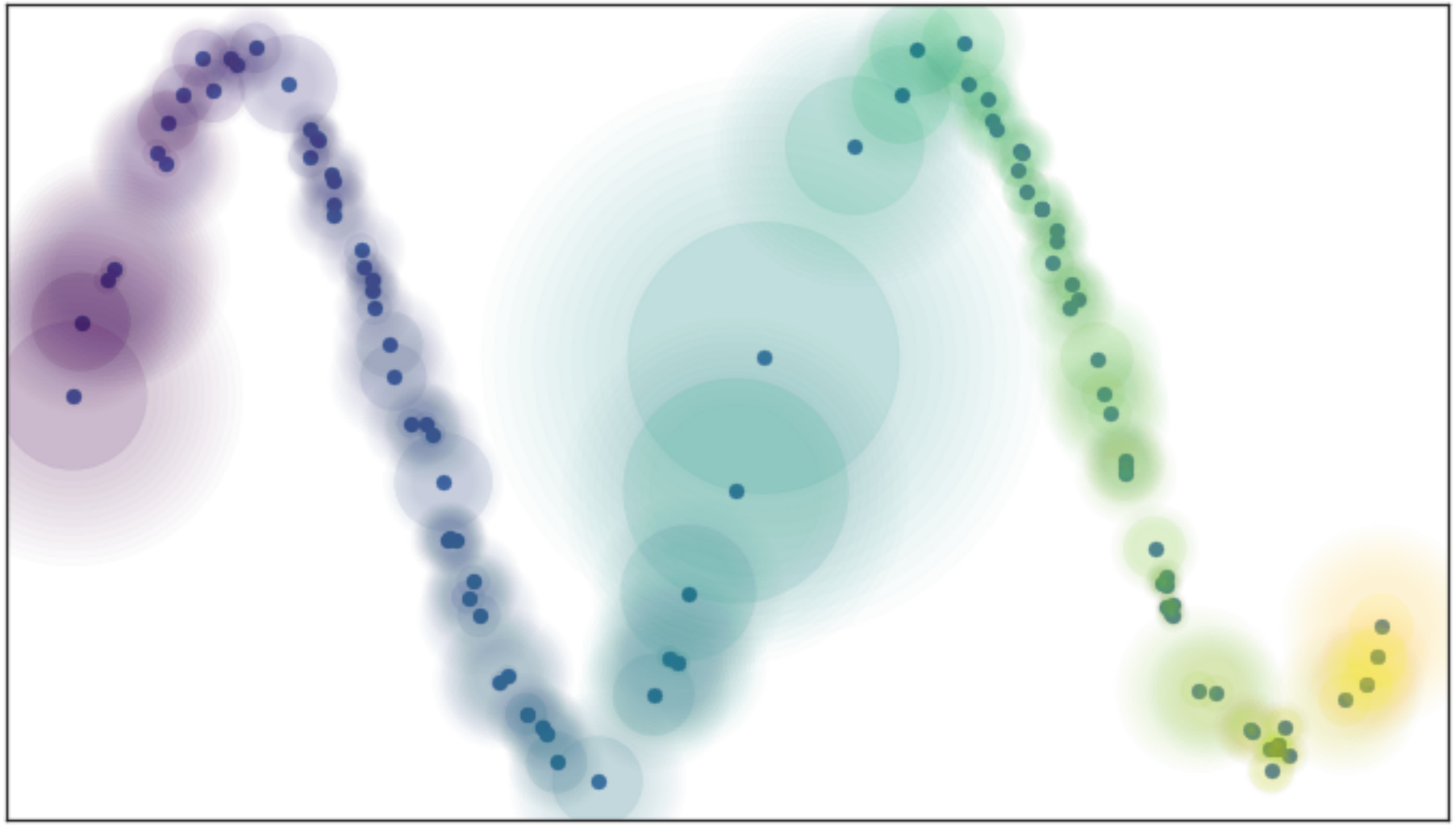
UMAP mit zwei Dimensionen



Flexible Radien finden auch Nachbarn für isolierte Punkte

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

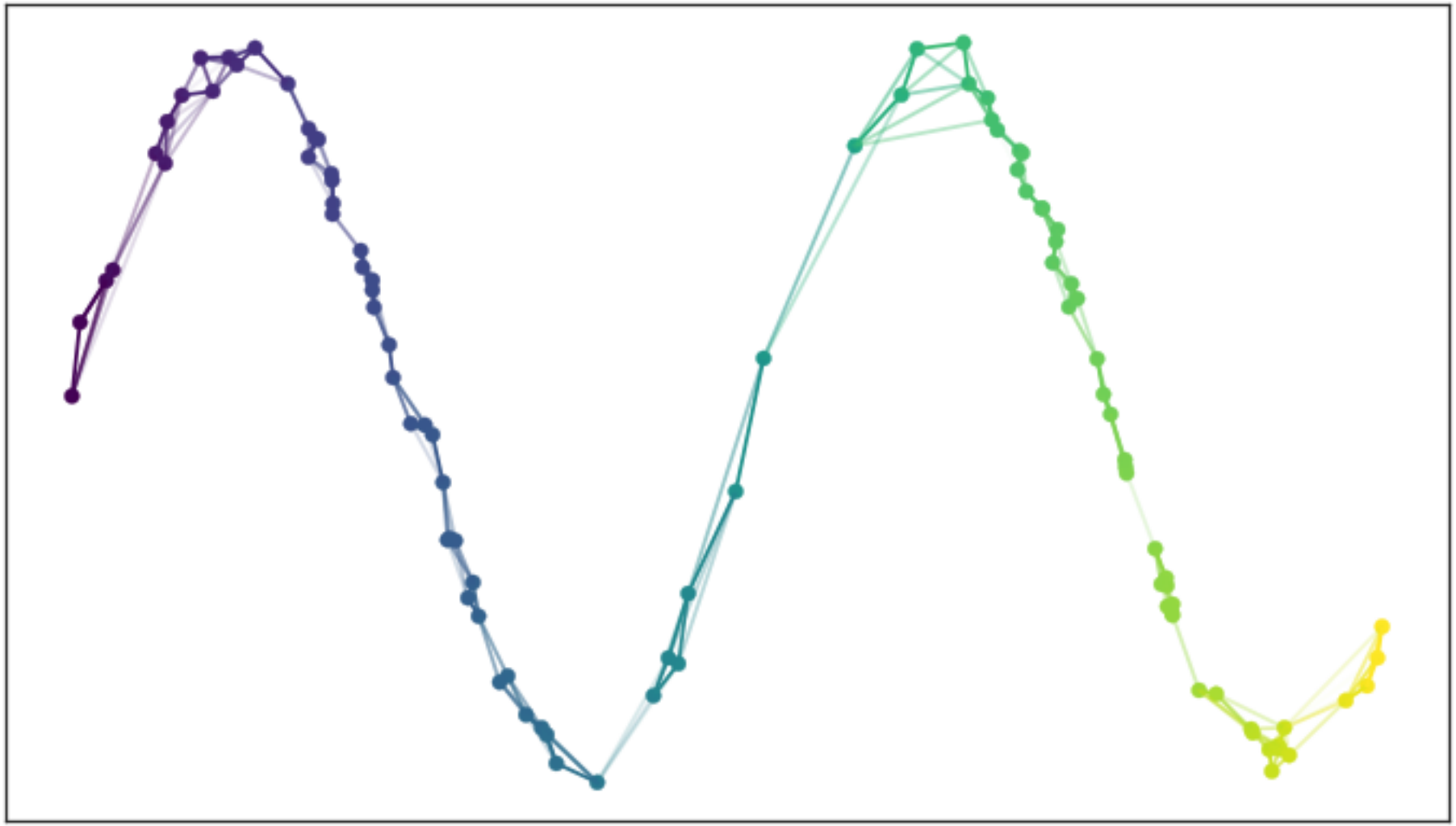
UMAP mit zwei Dimensionen



Eine Kombination eines harten Radius bis zum ersten Nachbarn und eines flexiblen drüber hinaus ist praktischer, weil...

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

UMAP mit zwei Dimensionen



...es uns erlaubt Wahrscheinlichkeiten zu berechnen, dass zwischen den Punkten eine Verbindung herrscht.

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

Dimensionsreduktion in UMAP

- Optimierungsalgorithmus für die “flexible” Abstandsmessung um den Platz in einem niedriger dimensionalen Raum zu nutzen.

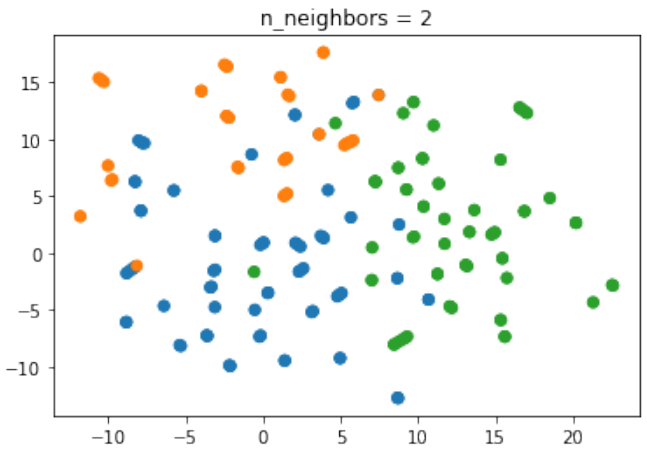
- Das Ergebnis der Optimierung ist abhängig von den Daten und hat eine Zufallskomponente.

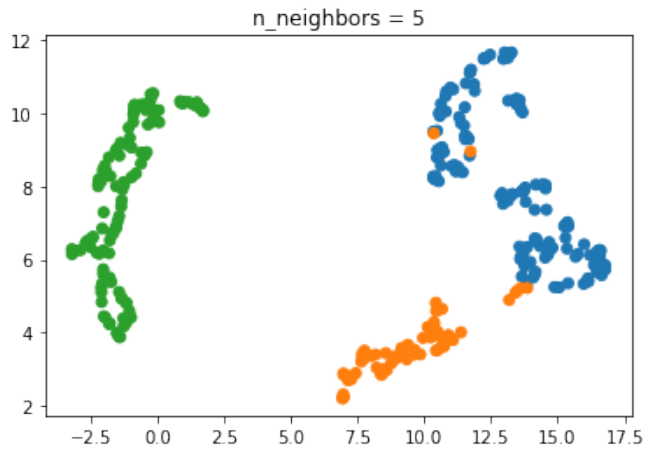
- Stellschrauben von UMAP sind…
  …die “Art” der Distanzmessung (metric)
  …nach wie vielen Nachbarn der Algorithmus sucht (n_neighbors)
  …wie stark Punkte aufeinander konzentriert werden dürfen (min_dist)

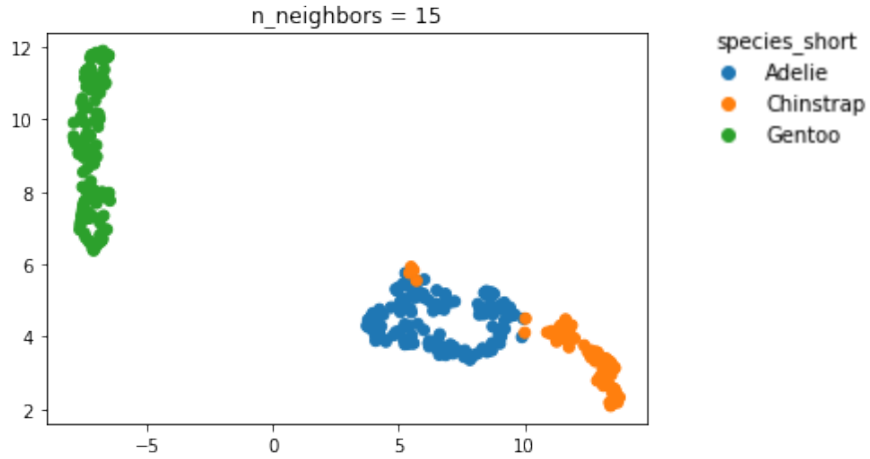
https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

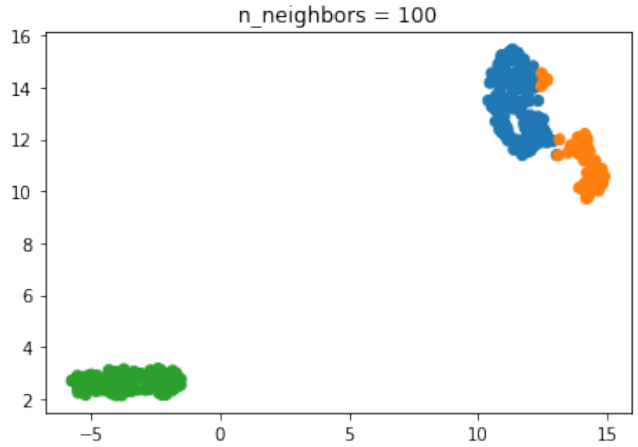
n_neighbors

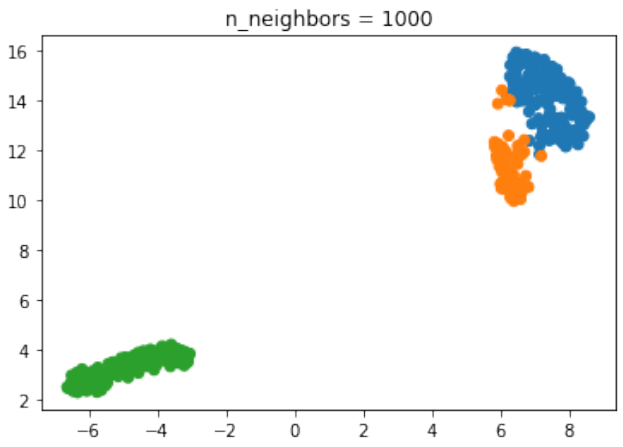
In [97]:
```
for n in (2, 5, 15, 100, 1000):
    reducer = umap.UMAP(n_neighbors=n)
    embedding = reducer.fit_transform(scaled_penguin_data)
    plt.scatter(
        embedding[:, 0],
        embedding[:, 1],
        c=[sns.color_palette()[x] for x in penguins.species_short.map({"Adelie":0, "Chinstrap":1, "Gentoo":2})]
        )
    plt.title('n_neighbors = {}'.format(n))
    plt.show()
```







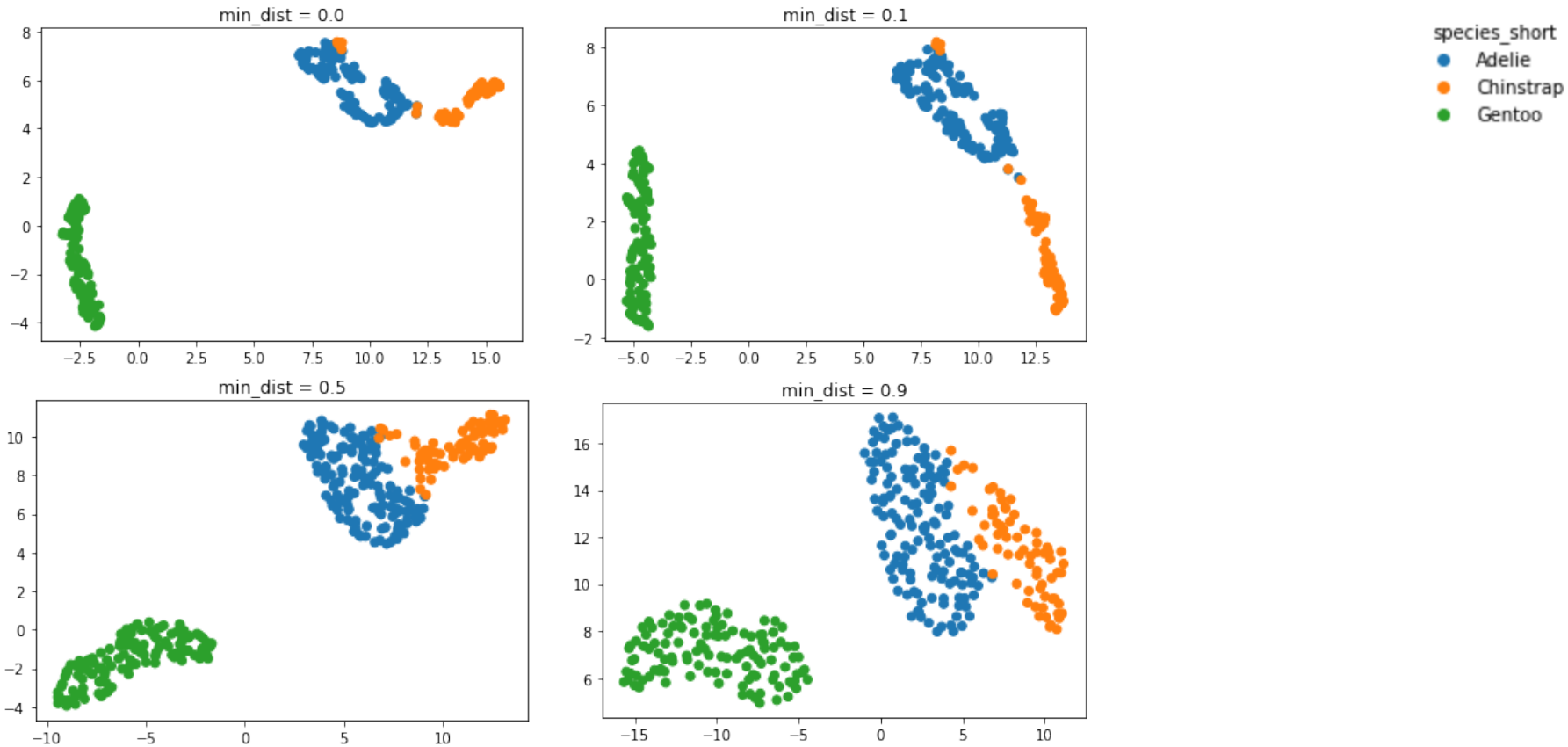





```
UserWarning: n_neighbors is larger than the dataset size
```

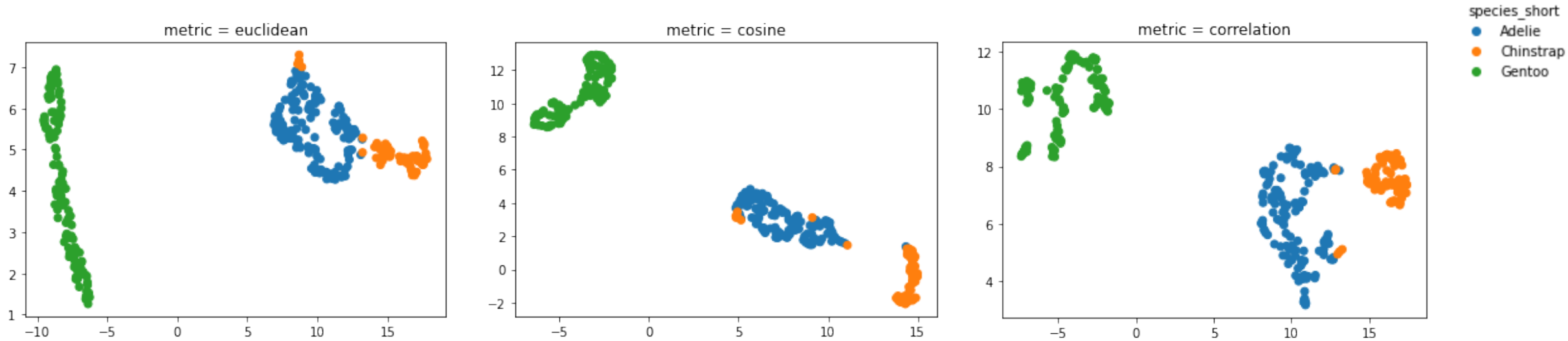
min_dist

In [19]:
```
for d in (0.0, 0.1, 0.5, 0.9):
    reducer = umap.UMAP(min_dist=d)
    embedding = reducer.fit_transform(scaled_penguin_data)
    plt.scatter(
        embedding[:, 0],
        embedding[:, 1],
        c=[sns.color_palette()[x] for x in penguins.species_short.map({"Adelie":0, "Chinstrap":1, "Gentoo":2})]
        )
    plt.title('min_dist = {}'.format(d))
    plt.show()
```



metric

In [69]:
```
for m in ("euclidean","cosine","correlation"):
    reducer = umap.UMAP(metric=m)
    embedding = reducer.fit_transform(scaled_penguin_data)
    plt.scatter(
        embedding[:, 0],
        embedding[:, 1],
        c=[sns.color_palette()[x] for x in penguins.species_short.map({"Adelie":0, "Chinstrap":1, "Gentoo":2})]
        )
    plt.title('metric = {}'.format(m))
    plt.show()
```



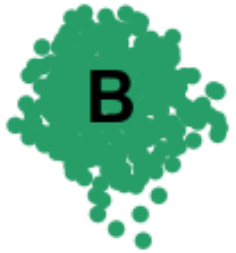
Woran man bei UMAP auch denken muss:

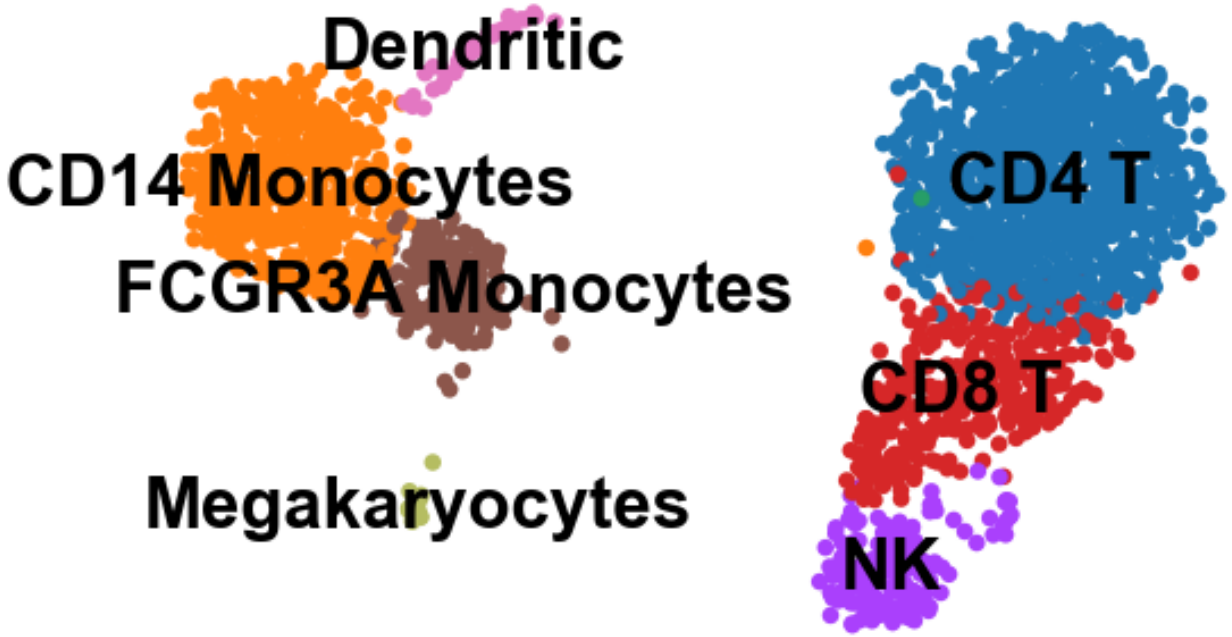
- Viele Parameter erlauben viel Spielraum die Daten “anzupassen”
- Gefahr die visuelle “Distanz” überzuinterpretieren
- Die Erhaltung der Datenstruktur ist nach wie vor ein Streitpunkt

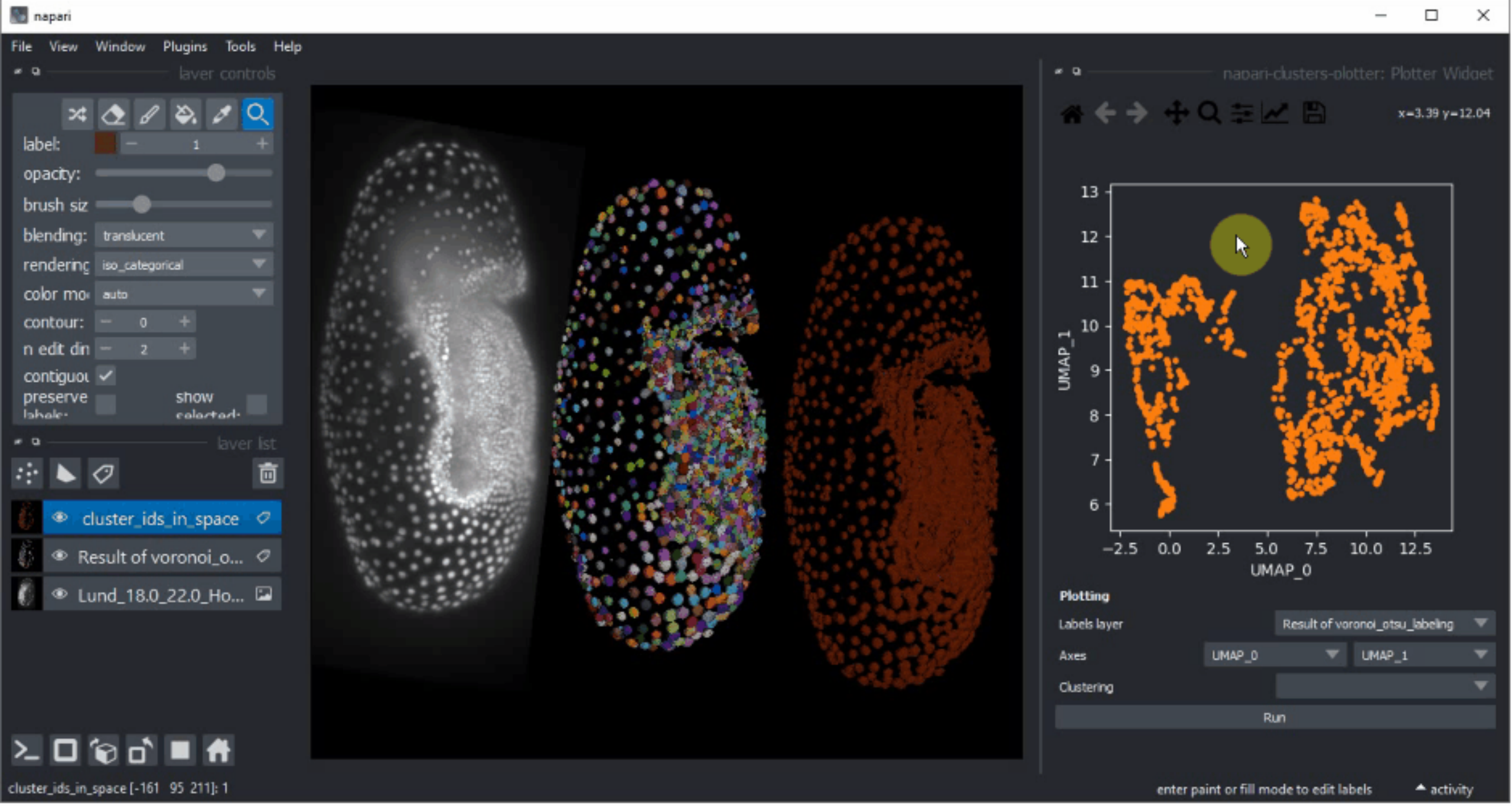
https://www.nature.com/articles/s41587-020-00809-z

Anwendungen von UMAP

Ready to go Tutorials:
scRNA-Seq tutorial in Python: https://github.com/theislab/single-cell-tutorial
scRNA-Seq blood analysis in Python: https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html
scRNA-Seq blood analysis in R: https://satijalab.org/seurat/articles/pbmc3k_tutorial.html







@haesleinhuepf

https://www.napari-hub.org/plugins/napari-clusters-plotter

Quellen und Material:

Tutorial: https://umap-learn.readthedocs.io/en/latest/
Paper: https://arxiv.org/abs/1802.03426
scRNA-Seq tutorial in Python: https://github.com/theislab/single-cell-tutorial
blood analysis in Python: https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html
blood analysis in R: https://satijalab.org/seurat/articles/pbmc3k_tutorial.html

Jupyter notebook (mit Bonusmaterial) & Präsentation:

https://github.com/arpoe/UMAP/