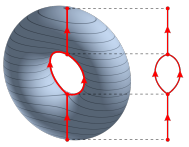
Exercise 1 *Kepler Mapper - Getting Started*
Revisit the example from the lecture and play around with different filter functions, noise-levels and covers.

```
# Initialize Mapper
import kmapper as km
mapper = km.KeplerMapper()

# Import sample data (2 disjoint circles)
from sklearn import datasets
data, labels = datasets.make_circles(n_samples=5000, noise=0.03, factor=0.3)

# Fit to and transform the data
projected_data = mapper.fit_transform(data, projection=[0,1])

# Create dictionary called 'graph' with nodes, edges and meta information
graph = mapper.map(projected_data, data)

# Visualize it
mapper.visualize(graph, path_html="getting_started.html",
                 title="make_circles(n_samples=5000,noise=0.03,factor=0.3)")
```

Exercise 2 *The Reeb Graph*
Let X be a topological space and $f : X \to \mathbb{R}$ a continuous function. The Reeb Graph $R_f(X)$ is the space obtained by identifying $x, y \in f^{-1}(c)$, whenever they lie in the same connected component of the level set $f^{-1}(c)$.



(a) Determine the Reeb Graph of the standard embedding of a torus $T = S^1 \times S^1$ in $\mathbb{R}^3$ for different filter functions f. (E.g. rotate the torus before projecting to the z-axis, or come up with your own filter).

(b) Use `tadasets.torus()` to construct the Mapper graph of a noisy torus for several predefined and/or custom filters.

(c) Add a puncture to the torus by removing (enough) points in a certain area. How does this affect the Reeb and Mapper graphs?

Exercise 3 *Kepler Mapper - Digits Datasets*
In this exercise we will investigate a dataset of digits and make ourselves familiar with some advanced Kepler Mapper functionalities.
for copyable code see https://kepler-mapper.scikit-tda.org/generated/gallery/plot_digits.html

The dataset we will use is part of `sklearn.datasets` and can be accessed via:

```
from sklearn import datasets
import matplotlib.pyplot as plt

#Load the digits dataset
digits = datasets.load_digits()

#Display the first digit
plt.figure(1, figsize=(3, 3))
plt.imshow(digits.images[0], cmap=plt.cm.gray_r, interpolation='nearest')
plt.show()
```

Make yourself familiar with the data: Look at the data structure, plot the first few digits.

Now it's time to apply Kepler Mapper to the dataset. We use a t-SNE filter function with 2 components (reduces data to 2 dimensions) that is provided by `sklearn`.

```
import sklearn
import kmapper as km
mapper = km.KeplerMapper(verbose=2)

# Fit and transform data
projected_data = mapper.fit_transform(digits.data, projection=sklearn.manifold.TSNE())

# Create the graph (we cluster on the projected data and suffer projection loss)
graph = mapper.map(projected_data,
                   clusterer=sklearn.cluster.DBSCAN(eps=0.3, min_samples=15),
                   cover=km.Cover(35, 0.4))
```

We now create a visualization that will be written to html.

```
# Create the visualizations
print("Output graph examples to html" )
# Tooltips with image data for every cluster member
mapper.visualize(graph,
                 title="Handwritten digits Mapper",
                 path_html="digits_custom_tooltips.html",
                 )
```

We can also have a look at the data with the help of `matplotlib.pyplot`:

```
# Matplotlib examples
import matplotlib.pyplot as plt
km.draw_matplotlib(graph, layout="spring")
plt.show()
```

Exercise 4 *Kepler Mapper - The Breast Cancer Dataset*
The dataset used by Nicolau et al (2011)[1] is accessible at
https://www.kaggle.com/uciml/breast-cancer-wisconsin-data
Have a look at the data and experiment with it.

A view instructions, in particular two possible filter functions, can be found in the Kepler Mapper Gallery at
https://kepler-mapper.scikit-tda.org/generated/gallery/plot_breast_cancer.html

[1] https://www.pnas.org/content/108/17/7265