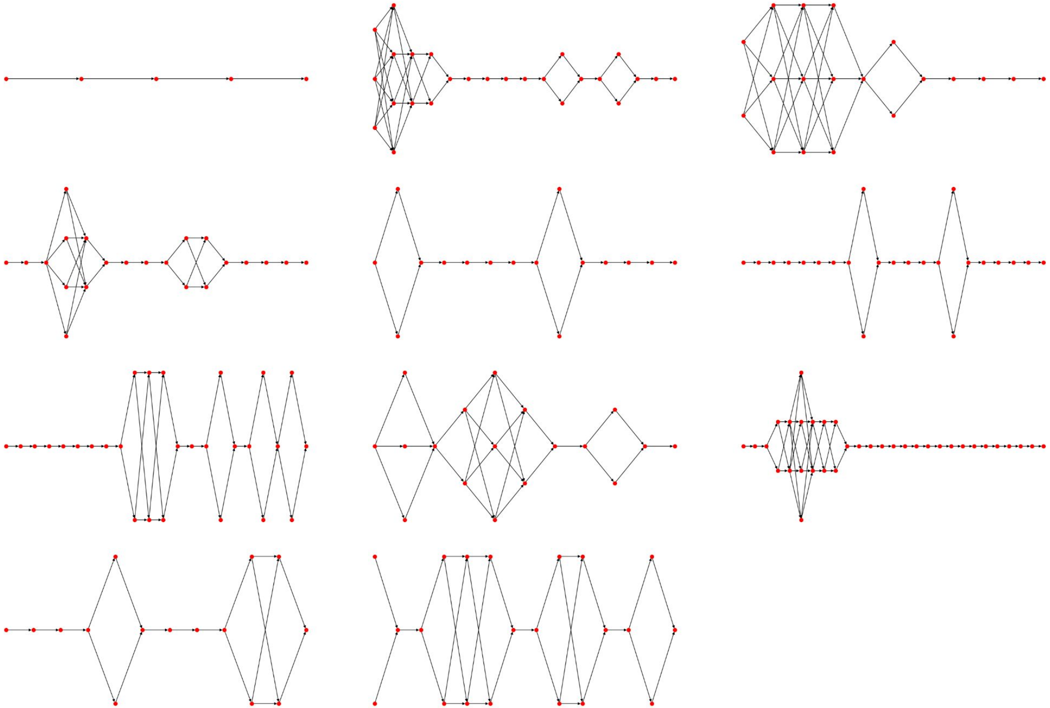
Geometric learning for quantitative analysis of stone tool reduction sequences

https://journals.openedition.org/palethnologie/1214



How to put a distance on a set of graphs?

The **quotient** distance w.r.t. to permutations of the nodes measures **structural** changes

It relies on a quadratic assignment problem (see scipy.optimize)

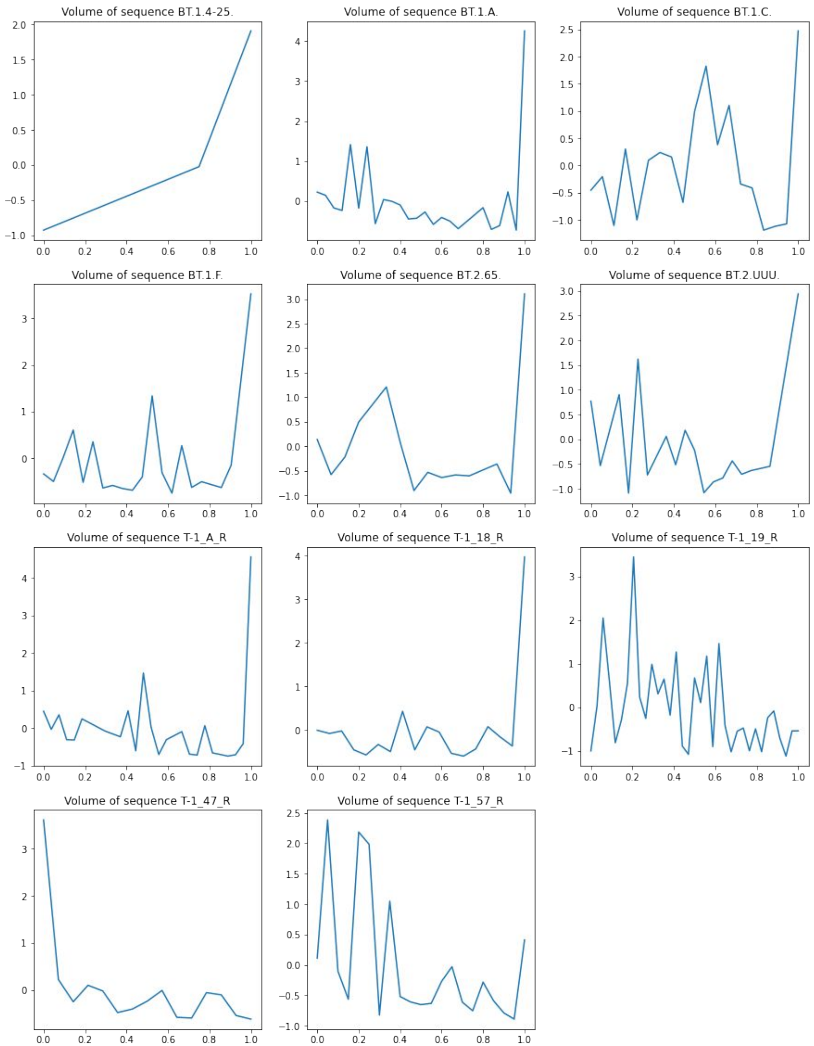
Measuring structural changes and changes of nodes (flakes) features **simultaneously** is a puzzle

Time series from graph

Randomly solve ambiguities to create a time series

NaN values in data -> fill with linear interpolation

Normalize data: sequence on interval [0,1] with mean 0, std 1 -> unitless



Metrics to compare time series

Different discretizations -> sample uniformly across interval

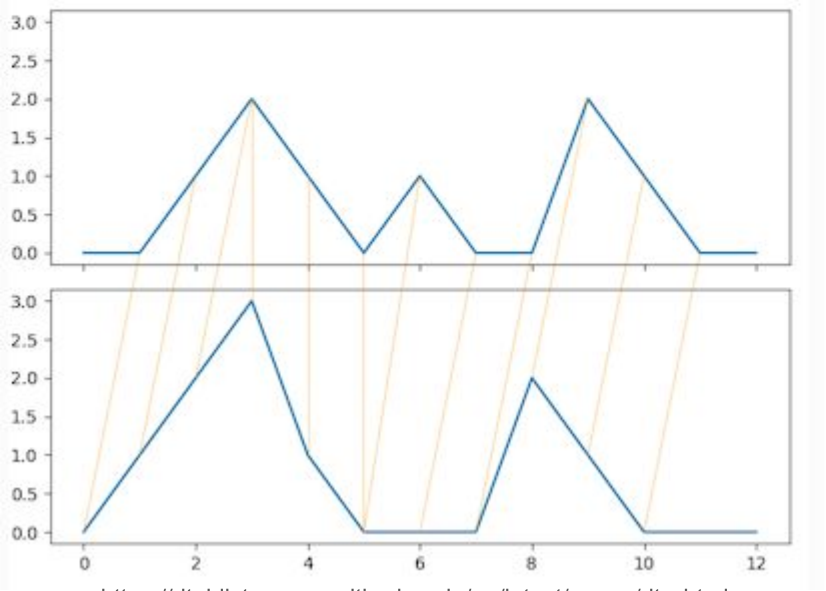
Naive: L2 distance

Dynamic Time Warping (DTW)

Not finished: Square Root Velocity (SRV)

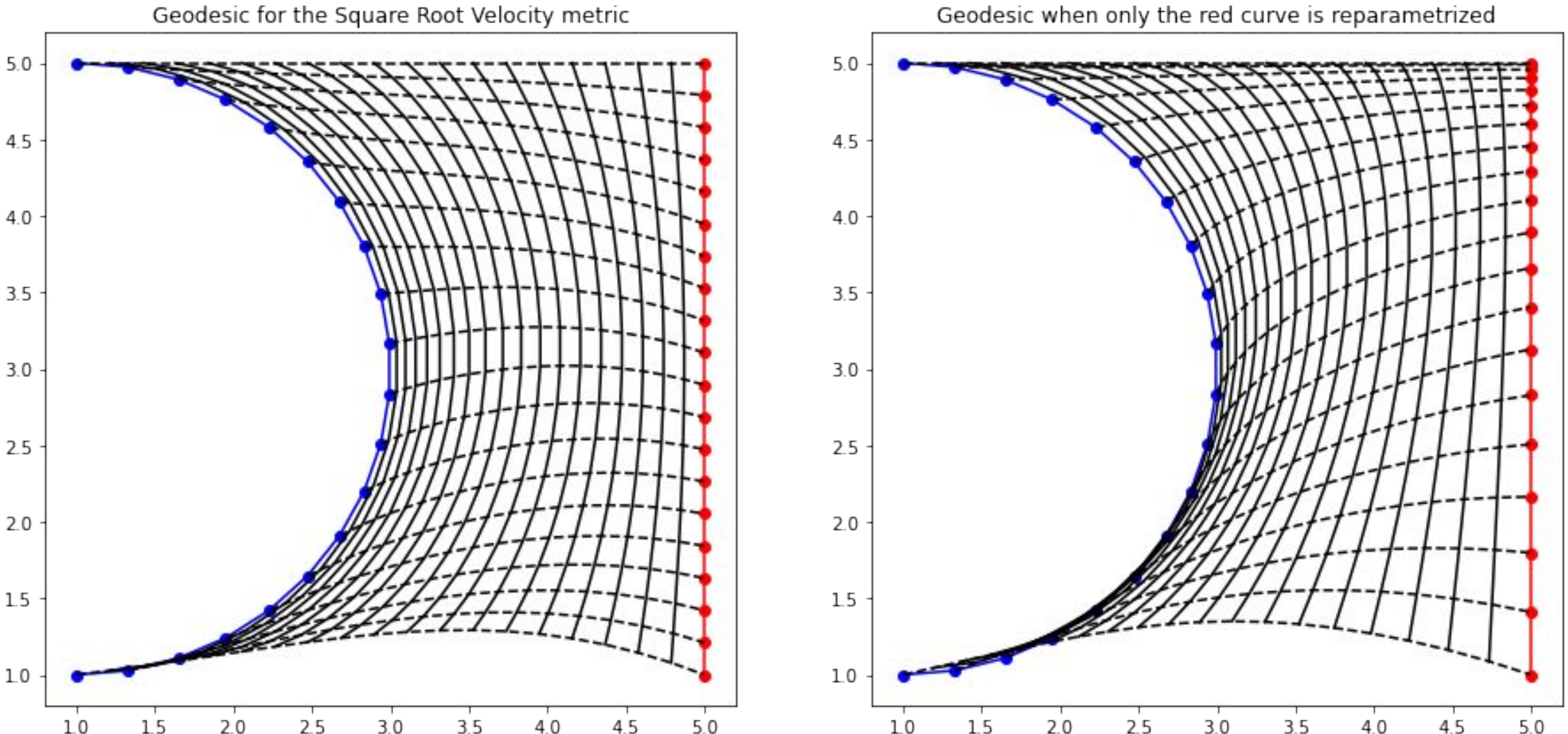
Could try more…

Dynamic Time Warping



https://dtaidistance.readthedocs.io/en/latest/usage/dtw.html

Square Root Velocity (SRV) metric



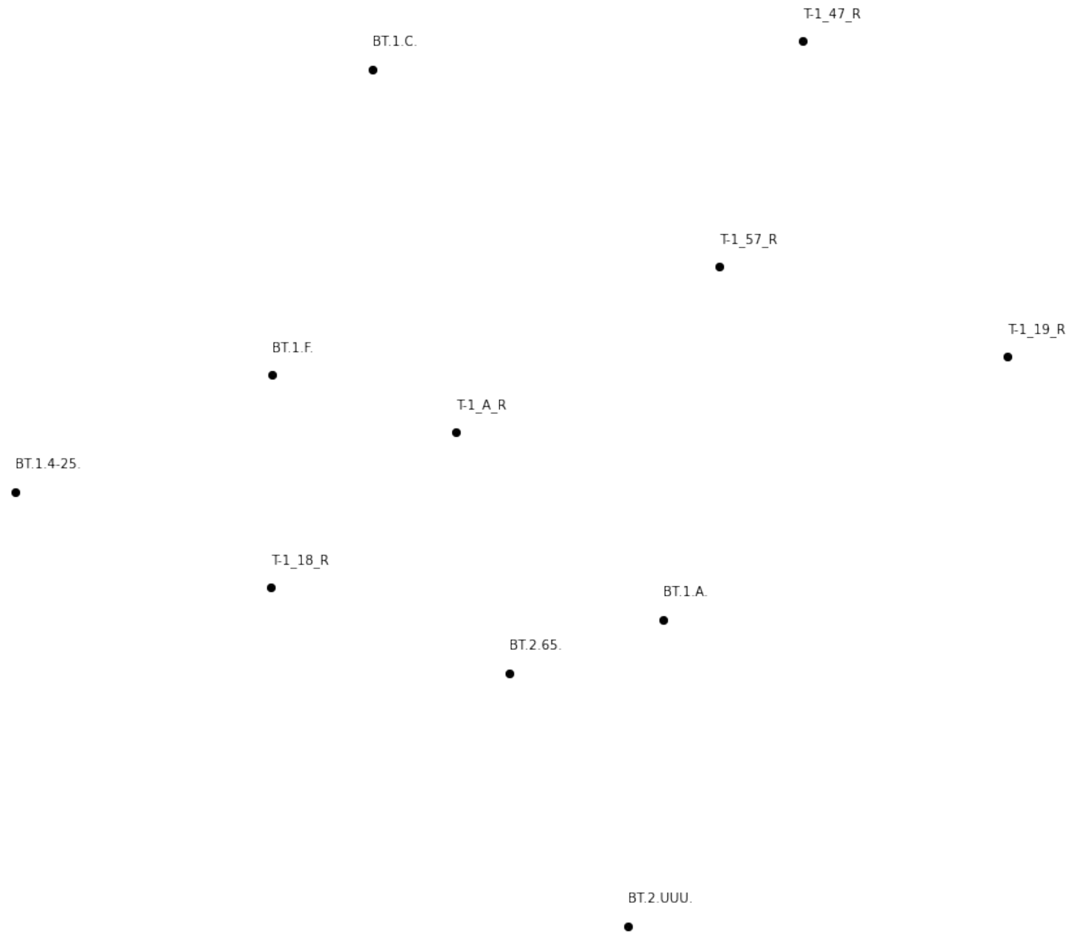
Visualize distance matrices

Calculated distances pairwise -> distance matrices

Different ways to visualize:

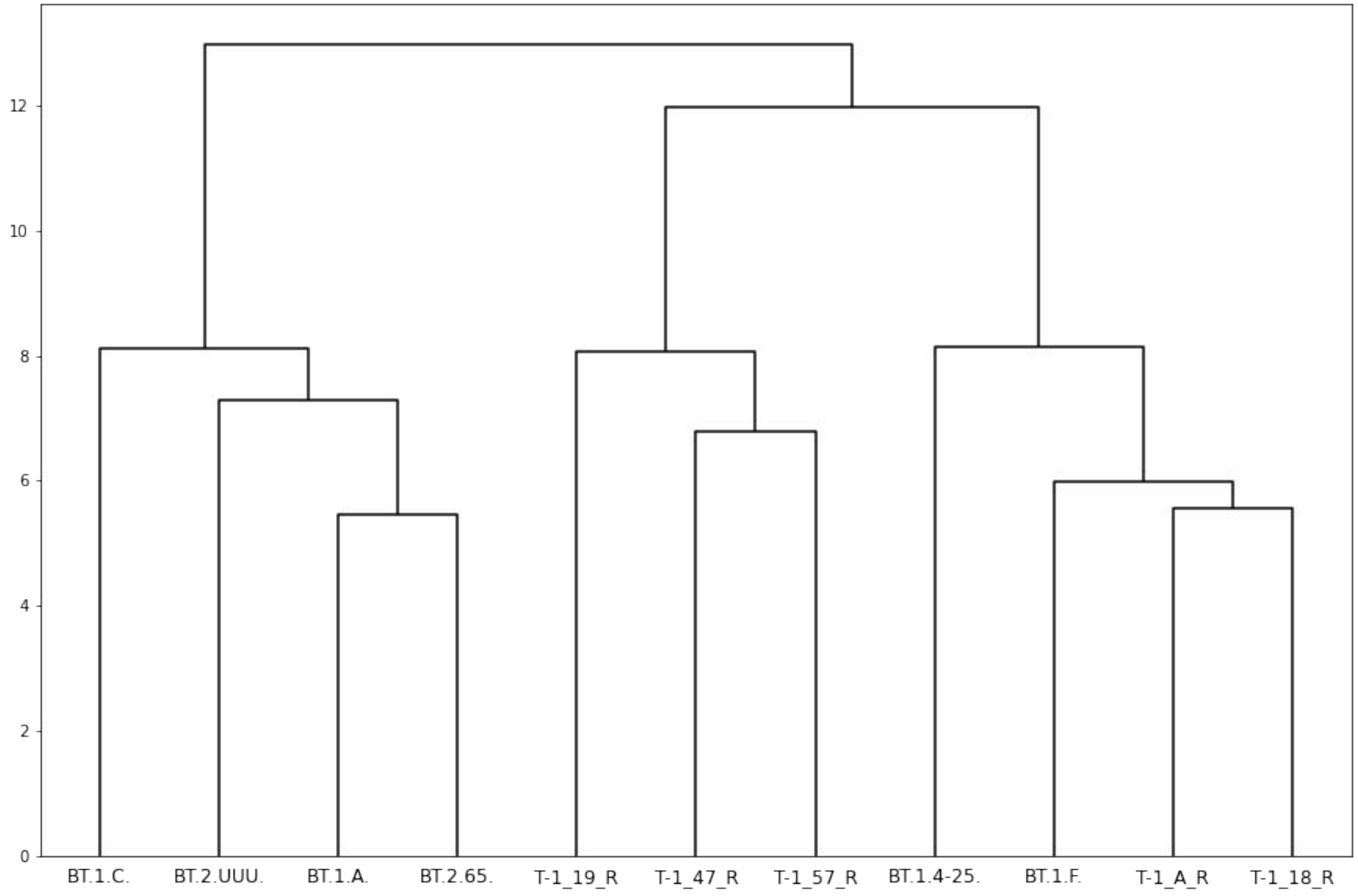
- MDS (Multidimensional scaling) from sklearn.manifold
- Diffusion maps
- Hierarchical clustering

MDS DTW


T-1_47_R
BT.1.C.
T-1_57_R
T-1_19_R
BT.1.F.
T-1_A_R
BT.1.4-25.
T-1_18_R
BT.1.A.
BT.2.65.
BT.2.UUU.

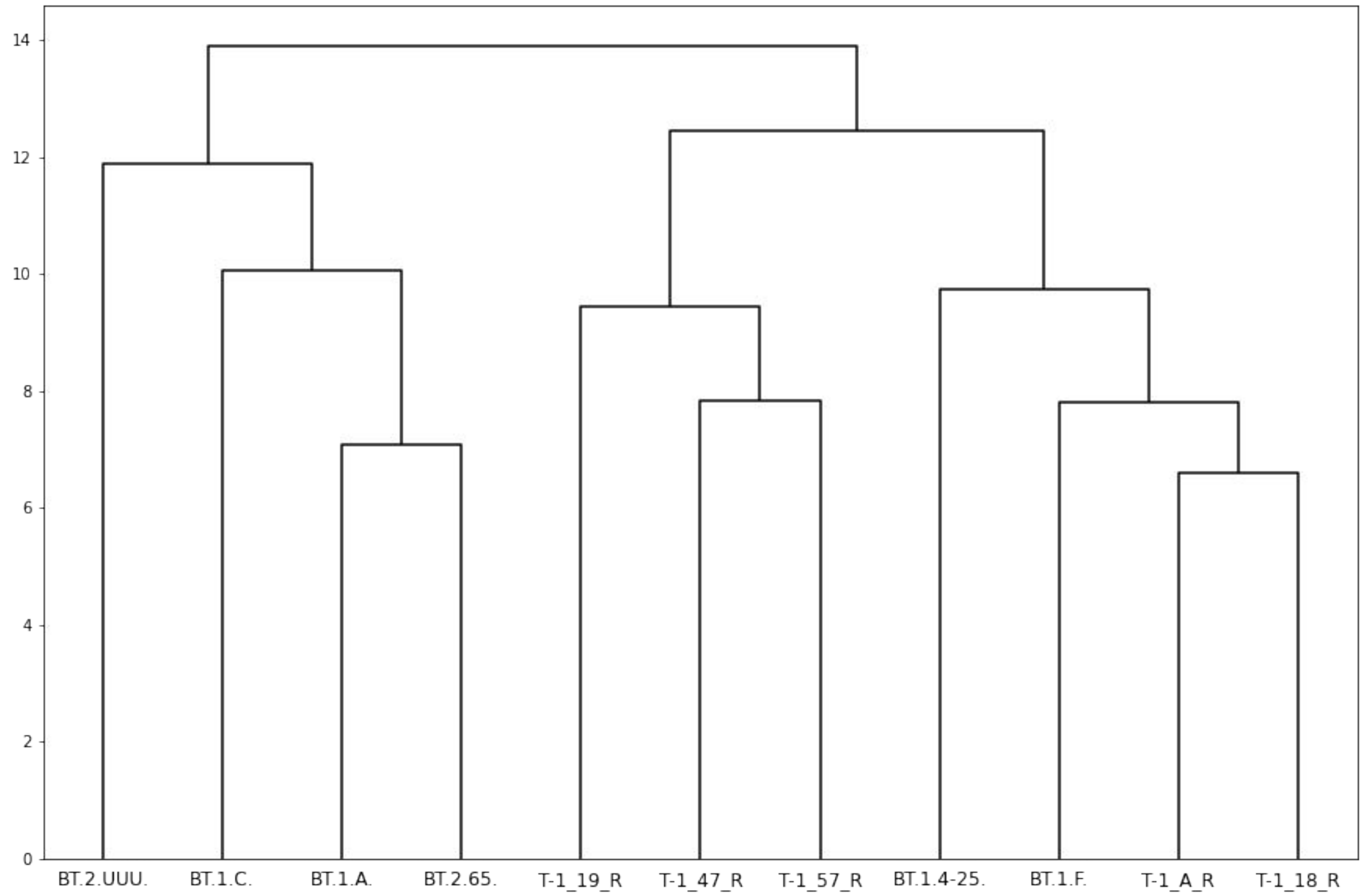
Diffusion DTW

Cluster DTW



12
10
8
6
4
2
0
BT.1.C.
BT.2.UUU.
BT.1.A.
BT.2.65.
T-1_19_R
T-1_47_R
T-1_57_R
BT.1.4-25.
BT.1.F.
T-1_A_R
T-1_18_R

Cluster L2



14
12
10
8
6
4
2
0
BT.2.UUU.
BT.1.C.
BT.1.A.
BT.2.65.
T-1_19_R
T-1_47_R
T-1_57_R
BT.1.4-25.
BT.1.F.
T-1_A_R
T-1_18_R

How to include the features while keeping the actual graphs?

- functionwise correspondence maps for graphs:
  - $\min_C ||C\phi_1 A - \phi_2 B||$
  - refine C
- compute latent functions
- compute inner products on the space of latent functions
- use them for clustering