Progress update

Philip Hartout

April 15, 2022

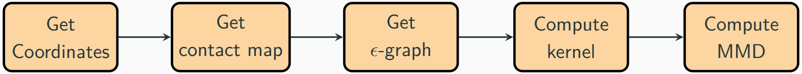
Overview

Measures	MMD				
Kernels	Graph Kernels	Vector Kernel	TDA Kernels	Kernel Composition	
Descriptors	Graph Descriptors	TDA Descriptors	Sequence Embeddings	Protein descriptors	
Graph	ε graphs	k-NN graphs			Graph Perturbations
Parsing	Point Clouds	Sequence	Mutations	Geometric Perturbations	Gaussian Noise
Files	PDB Files				

Green: perturbations

Orange: modules

Composable transformations & using `sklearn` API standards sensibly

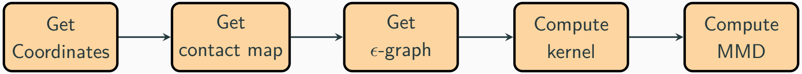



```python
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins = base_feature_pipeline.fit_transform(paths_to_pdb_files)

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```

Composable transformations & using `sklearn` API standards sensibly




```
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins = base_feature_pipeline.fit_transform(paths_to_pdb_files)

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```

What if we now want to add noise?

Composable transformations & using `sklearn` API standards sensibly

Get Coordinates → Add gaussian noise → Get contact map → Get ϵ-graph → Compute kernel → Compute MMD

```python
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins = base_feature_pipeline.fit_transform(paths_to_pdb_files)

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```

```python
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        (
            "add gaussian noise",
            GaussianNoise(
                random_seed=42, noise_mean=0, noise_variance=10, n_jobs=12,
            ),
        ),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins_perturbed = base_feature_pipeline.fit_transform(paths_to_pdb_files)

graphs = load_graphs(proteins, graph_type="eps_graph")
graphs_perturbed = load_graphs(proteins_perturbed, graph_type="eps_graph")

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```

Reusable Components

But I want results!

MMD experiments informs the best representation to use for proteins

Get Coordinates → Add gaussian noise → Get contact map → Get ϵ-graph → Compute kernel → Compute MMD

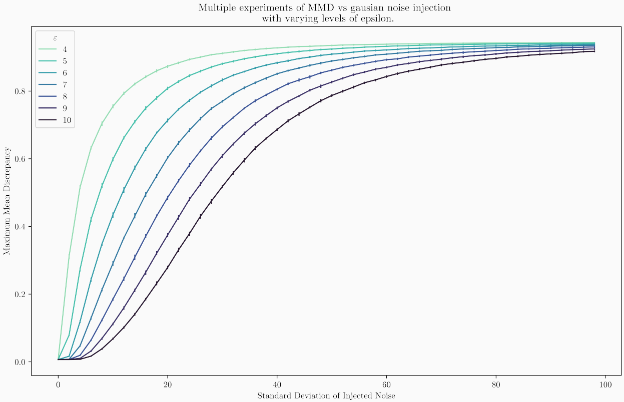
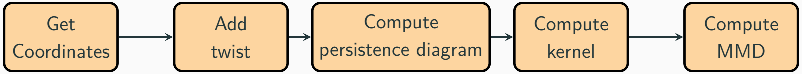


Figure 1: MMD as gaussian noise is added. Each bar is the 100% confidence interval over 10 runs.

TDA captures morphological perturbations of proteins



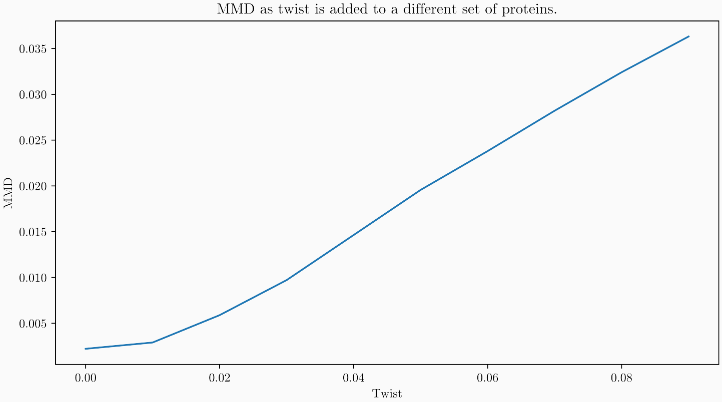


Figure 2: MMD value as twist is progressively added to a separate set of proteins.

1-sentence takeway

Parametrize your MMD sensibly for your protein generative model.

Backup slides

Detailed breakdown of modules

Point clouds:

- Granularity can be set to α-Carbon, β-Carbon, entire backbone or all-atom setting.

Graph Descriptors:

- Degree Histogram
- Clustering Histogram
- Laplacian spectrum

Topological Descriptors

- Persistence diagrams
- Persistence landscape
- Persistence image
- Betti Curves

Sequence Embeddings (ESM, different sizes)

Protein Descriptors

- Ramachandran angles
- Interactomic clashes

Graph Kernels (Weisfeiler-Lehman Kernel)
Vector kernels (Linear, Gaussian)
TDA Kernel (Persistence Fisher Kernel)
Kernel composition ($\times$, $+$)
MMD (Squared, biased)

Perturbations
Graph level (rewire, add/remove edge)
Point cloud perturbations (twist, taper, shear)
Protein perturbations (mutate)
All modules work on multiple proteins simultaneously

The admin stuff

Message through slack for follow-up Qs

Extensive GitHub documentation & up-to-date codebase at

https://github.com/pjhartout/msc_thesis

Private repo a.t.m., message me to request access.