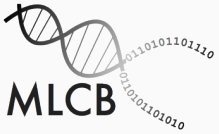
Building a library to evaluate generative protein models

M.Sc. Thesis Midterm Presentation

Philip Hartout

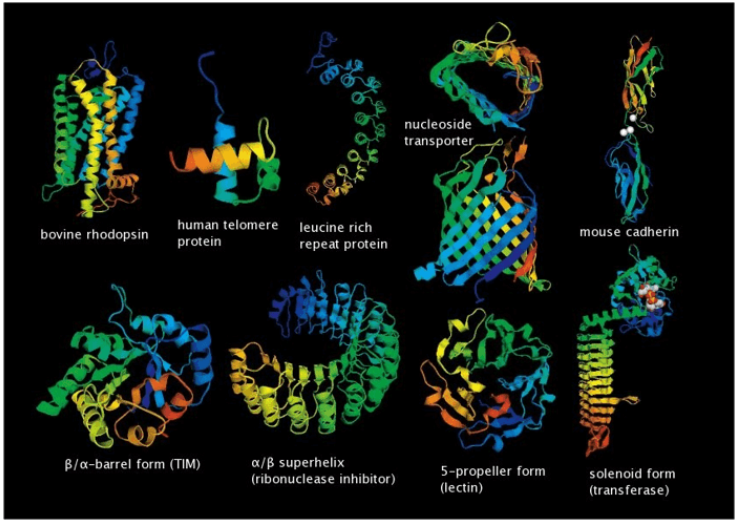
May 11, 2022



D BSSE

ETH zürich

Introduction



Proteins are diverse.

Support all functions for life.

Generative Protein Modelling

Proteins

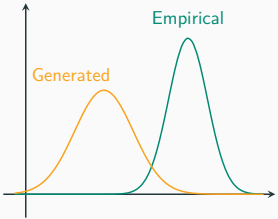
- well-defined (sequence)
- large databases

Generative Models

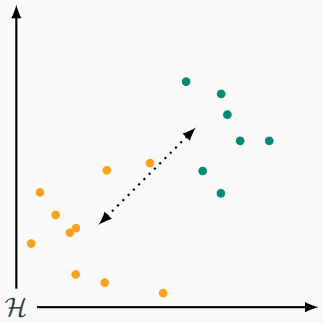
- capture $P(X)$
- generate samples following $P(X)$

Evaluation Problem

- Are the generated and empirical data distributions the same?



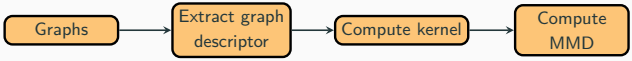
Maximum Mean Discrepancy (MMD)



MMD captures the distance between 2 sets on *any* RKHS $\mathcal{H}$.
Can be used in kernelized two-sample test.

Maximum Mean Discrepancy (MMD) – continued

Currently accepted method to evaluate generative GNNs.



It's possible to leverage decades of kernel research!

Both a blessing and a curse:

Blessing Flexibility, Computation on multiple representations

Curse Instability (see O'Bray et al. [2021]), hyperparameter tuning.

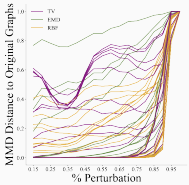


Figure 1: MMD computed from a clustering coefficient on synthetic graphs. TV: total variation kernel, RBF: radial basis function, EMD: earth mover's distance.

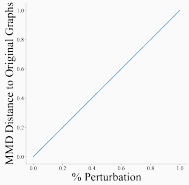


Figure 2: Ideal MMD behavior.

– Thesis Goal –

Building a library to evaluate generative protein models

Experimental Setup – Perturbations to Assess MMD

1. 10 pairs of 100 proteins from Alphafold DB (easy to work with)
2. Apply perturbation to 1 of the datasets in each pair.
3. Calculate MMD

Gaussian noise

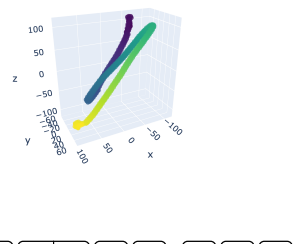


Figure 3: Adding Gaussian Noise. Color-coded according to the index.

Twist

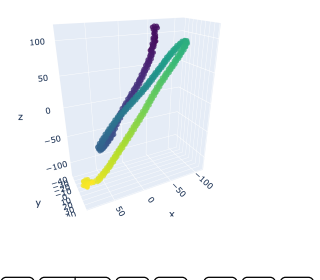
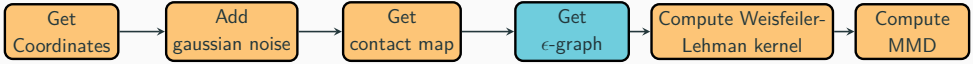
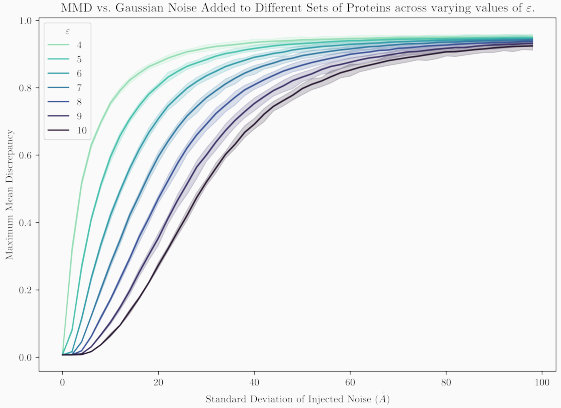


Figure 4: Adding Twist. Color-coded according to the index.

Experiment 1 – Gaussian Noise





Each curve: different ε.

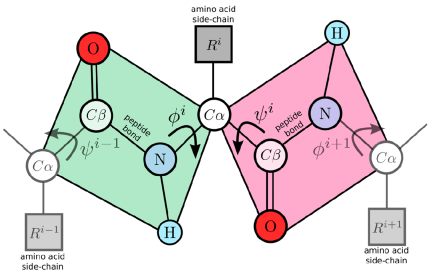
2 sources of variance:

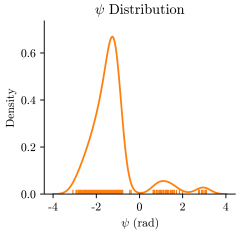
- Data
- Noise

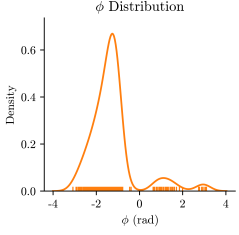
Conclusions

1. MMD is stable using the Weisfeiler-Lehman kernel
2. Choice of representation influences MMD

Experiment 2 Preliminaries 1/2 - Dihedral Angles



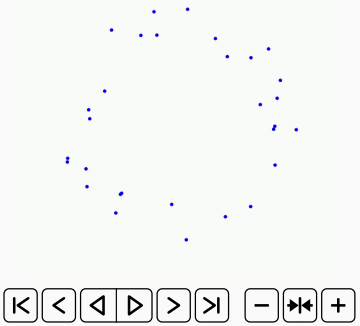


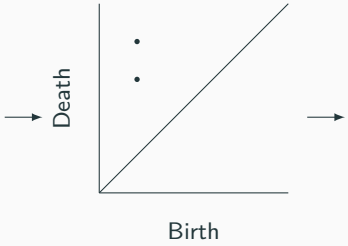


Use in MMD

- Concatenate
- Compute kernel (Gaussian, Linear, ...)

Experiment 2 Preliminaries 2/2 -Topological Data Analysis

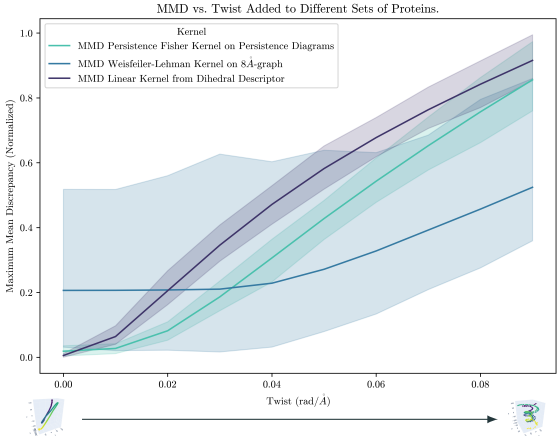




Kernels operate directly on persistence diagrams, e.g. Le and Yamada [2018].

Figure 6: Vietoris-Rips filtrations captures connected components, cycles and holes by varying ε.

Experiment 2 – Twist



1 source of variance:

- Data

Conclusions

1. TDA behaves very well, computationally complex
2. Dihedral descriptor behave well, fast to compute
3. Weisfeiler-Lehman kernel does not capture global shape changes

Outlook – Next steps

More realistic perturbations.

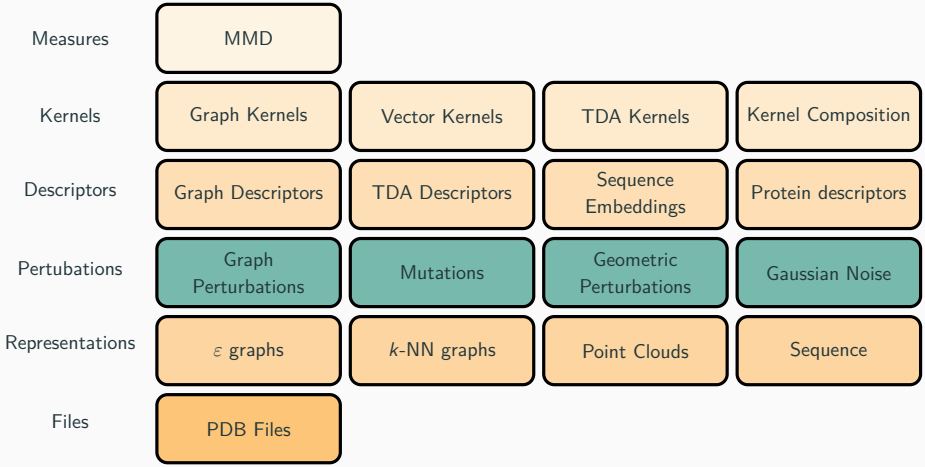
- Mutations (for engineering)

Data & Failure modes

- MMD with different protein families
- Mode dropping/collapse (tweak mix of each protein families?)
- Copy problem (w/ graph edit distance derivative?)

- Systematic assessment of kernels & hyperparameters
- Devise strategies to recommend best pipeline(s).

Questions - Overview

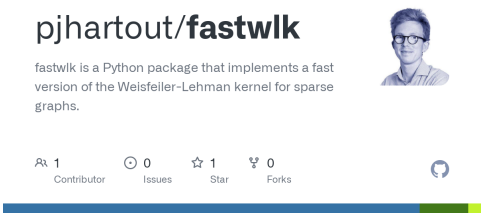


References

Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. *arXiv preprint arXiv:2106.01098*, 2021.

Tam Le and Makoto Yamada. Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. *Advances in Neural Information Processing Systems*, 31, 2018.

Spin-off project: implementing `fastwlk`



https://github.com/pjhartout/fastwlk

Three-pronged approach to decrease Weisfeiler-Lehman computation time for sparse graphs:

1. Compute W-L hashes independently for each graph
2. Distribute dot product
3. For each pair, only compute set of overlapping colors. Stems from observations that large graphs often only have a small set of unique common labels.

Maximum Mean Discrepancy (MMD)

$$\mathrm{MMD}(X, Y) := \frac{1}{n^2} \sum_{i,j=1}^{n} k(x_i, x_j) + \frac{1}{m^2} \sum_{i,j=1}^{n} k(y_i, y_j) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(x_i, y_j)$$

where:

- $\mathcal{X}$ is some non-empty set.
- $x_i, x_j \subseteq \mathcal{X}$, n is the number of samples in $\mathbf{x}$;
- $y_i, y_j \subseteq \mathcal{X}$, m is the number of samples in $\mathbf{y}$;
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a valid kernel.

MMD captures the distance between 2 sets on *any* RKHS $\mathcal{H}$.

API

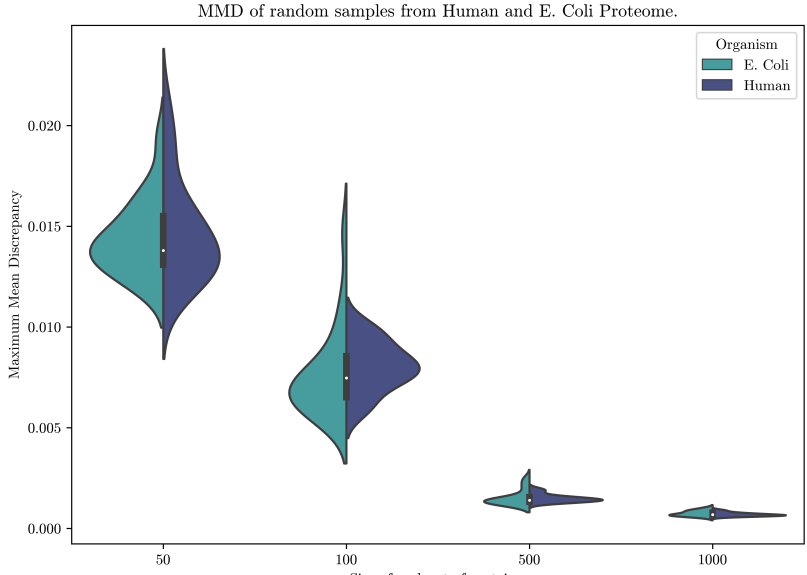
```
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        (
            "add gaussian noise",
            GaussianNoise(
                random_seed=42, noise_mean=0, noise_variance=10, n_jobs=12,
            ),
        ),
        ("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)
```

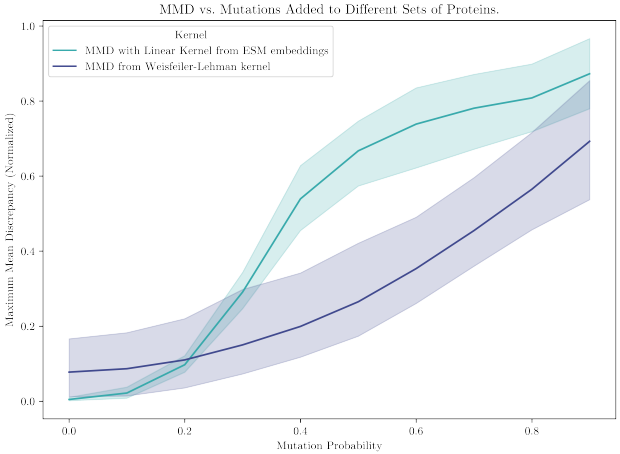
```
graphs = load_graphs(proteins, graph_type="eps_graph")
graphs_perturbed = load_graphs(proteins_perturbed, graph_type="eps_graph")

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

MMD with 8-Å-MMD with Weisfeiler-Lehman kernel



Experiment 3 – Mutate



2 sources of variance:

- Data
- Mutation seed

Conclusions

1. Weisfeiler-Lehman kernel captures changes but noisy
2. ESM captures changes.
3. Further study with lower mutation probabilities.