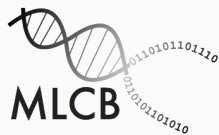


# Master's Thesis Midterm Presentation

---

Philip Hartout

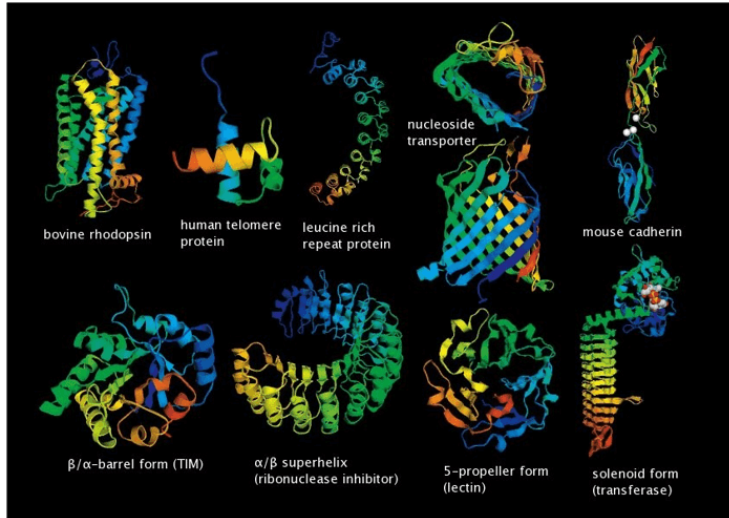
April 30, 2022



**DBSSE**

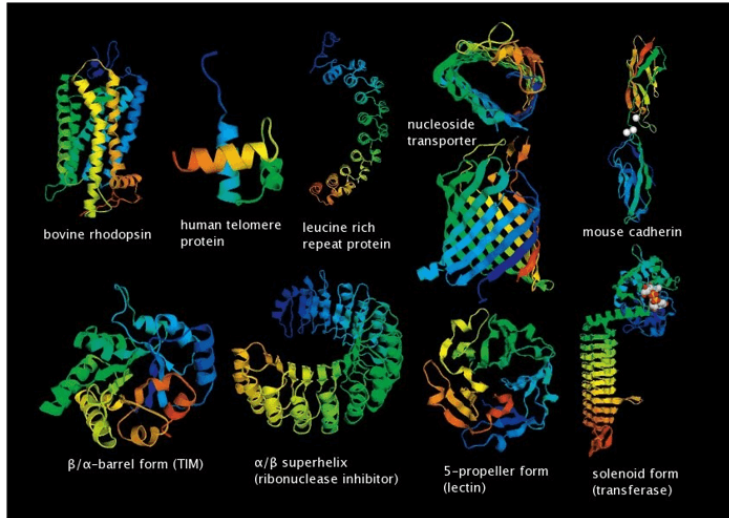
**ETH** zürich

# Introduction



Proteins are extremely diverse.

# Introduction



Proteins are extremely diverse.

They support all functions necessary for life.

# Generative Protein Modelling

Proteins are an interesting platform for generative modelling:

# Generative Protein Modelling

Proteins are an interesting platform for generative modelling:

- Proteins can be engineered (for therapeutic or industrial purposes)

# Generative Protein Modelling

Proteins are an interesting platform for generative modelling:

- Proteins can be engineered (for therapeutic or industrial purposes)
- Well-defined (boils down to a sequence or set of sequences)

# Generative Protein Modelling

Proteins are an interesting platform for generative modelling:

- Proteins can be engineered (for therapeutic or industrial purposes)
- Well-defined (boils down to a sequence or set of sequences)

**Generative Model** A *generative model* captures the probability distribution of  $P(X)$ , and is therefore capable of generating samples following  $P(X)$  given a random vector of inputs.

# Generative Protein Modelling

Proteins are an interesting platform for generative modelling:

- Proteins can be engineered (for therapeutic or industrial purposes)
- Well-defined (boils down to a sequence or set of sequences)

**Generative Model** A *generative model* captures the probability distribution of  $P(X)$ , and is therefore capable of generating samples following  $P(X)$  given a random vector of inputs.

We concern ourselves with the *evaluation problem* here.



# Generative Protein Modelling

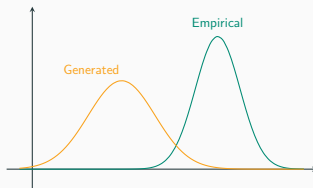
Proteins are an interesting platform for generative modelling:

- Proteins can be engineered (for therapeutic or industrial purposes)
- Well-defined (boils down to a sequence or set of sequences)

**Generative Model** A *generative model* captures the probability distribution of  $P(X)$ , and is therefore capable of generating samples following  $P(X)$  given a random vector of inputs.

We concern ourselves with the *evaluation problem* here.

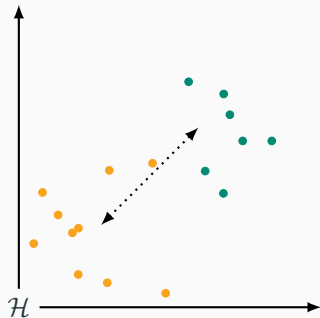
Given a set of proteins, how do we make sure that our model follows the same distribution?



**What makes a good protein?**

# Maximum Mean Discrepancy (MMD)

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



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  $\mathbf{y}$ ;
- $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a valid kernel.

MMD captures distances between 2 sets of structured data on any RKHS  $\mathcal{H}$ .

# Maximum Mean Discrepancy (MMD) – continued

Currently accepted method to evaluate GNNs. Advantages:

1. It's possible to leverage decades of kernel research!

Both a  **blessing**  and a  **curse** :

**Blessing** Flexibility, Computation on multiple representations

**Curse** Instability (see Leslie's ICLR work), hyperparameter tuning.

2. Possibility of statistical testing.
3. Possibility of leveraging multiple representations.

*Foreshadowing...*

**– Thesis Goal –**

**Build a library to evaluate protein generative models**

Files

PDB Files

# Overview

Representations

$\epsilon$  graphs

$k$ -NN graphs

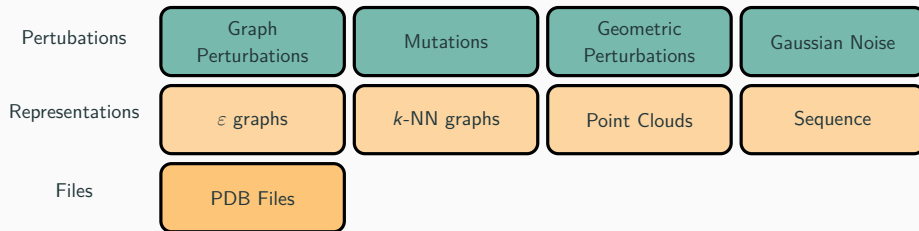
Point Clouds

Sequence

Files

PDB Files

# Overview





# Overview

Descriptors	Graph Descriptors	TDA Descriptors	Sequence Embeddings	Protein descriptors
Perturbations	Graph Perturbations	Mutations	Geometric Perturbations	Gaussian Noise
Representations	$\epsilon$ graphs	$k$ -NN graphs	Point Clouds	Sequence
Files	PDB Files			

# Overview

Kernels	Graph Kernels	Vector Kernels	TDA Kernels	Kernel Composition
Descriptors	Graph Descriptors	TDA Descriptors	Sequence Embeddings	Protein descriptors
Perturbations	Graph Perturbations	Mutations	Geometric Perturbations	Gaussian Noise
Representations	$\epsilon$ graphs	$k$ -NN graphs	Point Clouds	Sequence
Files	PDB Files			

# Overview

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

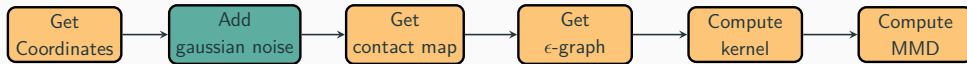
We will study the following settings:

1. Add gaussian noise to the point cloud and use graph kernels
2. Add twist to proteins to the point cloud and see the results on different kernels
3. Add mutations to proteins to the point cloud and see the results on different kernels

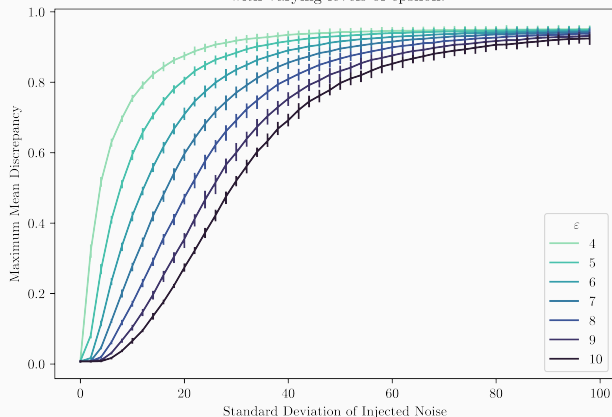
## Experiment 1 – Gaussian Noise



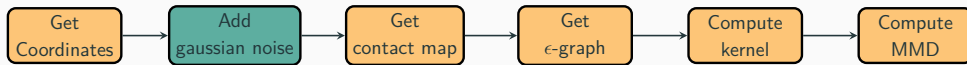
# Experiment 1 – Gaussian Noise



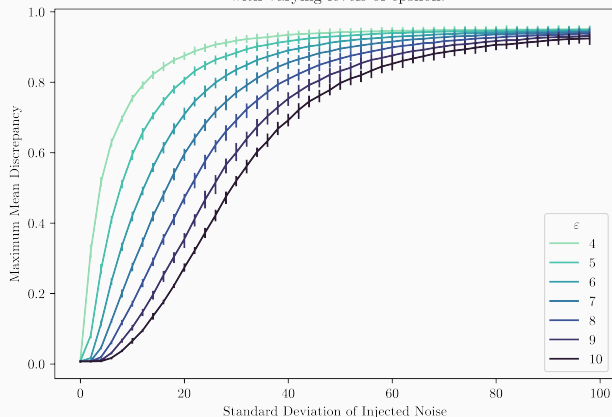
Multiple experiments of MMD vs gaussian noise injection with varying levels of epsilon.



# Experiment 1 – Gaussian Noise

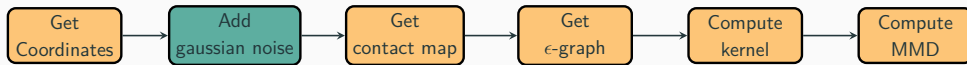


Multiple experiments of MMD vs gaussian noise injection with varying levels of epsilon.

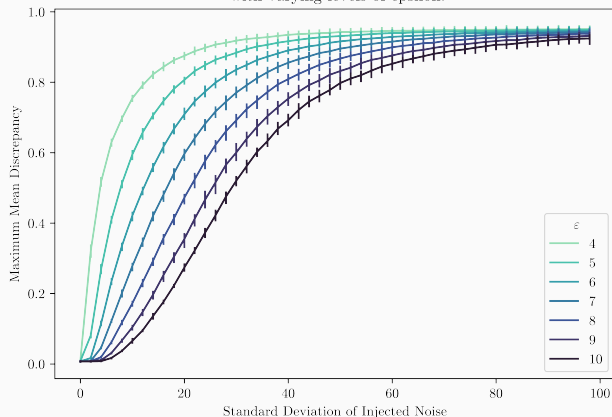


2 sources of variance:

# Experiment 1 – Gaussian Noise



Multiple experiments of MMD vs gaussian noise injection with varying levels of epsilon.



2 sources of variance:

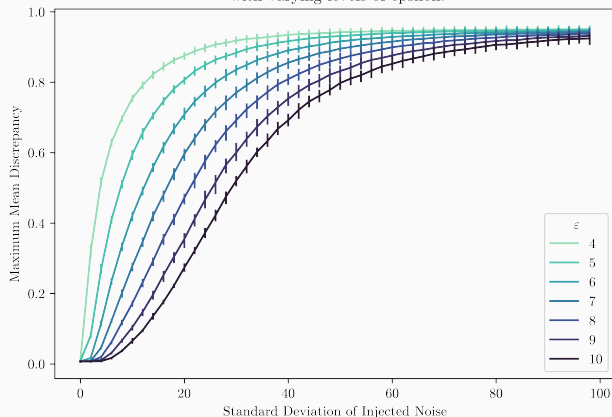
- Data



# Experiment 1 – Gaussian Noise



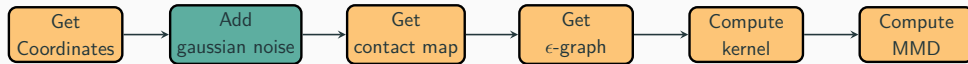
Multiple experiments of MMD vs gaussian noise injection with varying levels of epsilon.



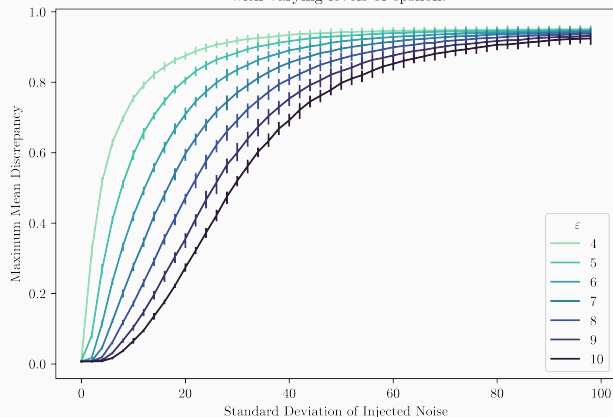
2 sources of variance:

- Data
- Noise

# Experiment 1 – Gaussian Noise



Multiple experiments of MMD vs gaussian noise injection with varying levels of epsilon.



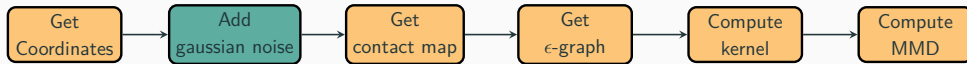
2 sources of variance:

- Data
- Noise

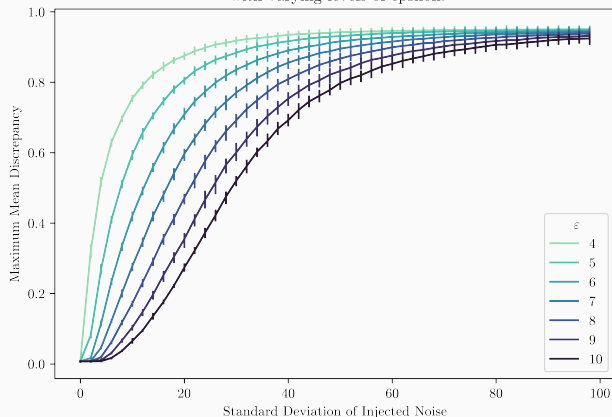
$n = 10$  runs

Conclusions

# Experiment 1 – Gaussian Noise



Multiple experiments of MMD vs gaussian noise injection with varying levels of epsilon.



2 sources of variance:

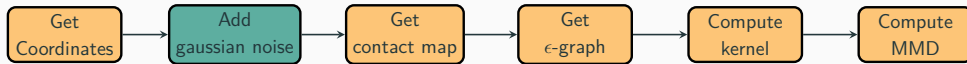
- Data
- Noise

$n = 10$  runs

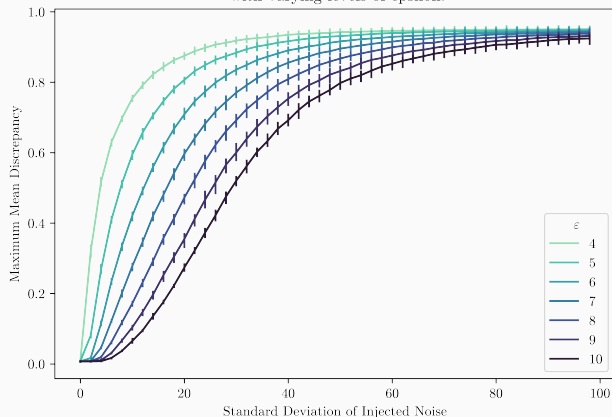
Conclusions

1. MMD is stable using the Weisfeiler-Lehman kernel

# Experiment 1 – Gaussian Noise



Multiple experiments of MMD vs gaussian noise injection with varying levels of epsilon.



2 sources of variance:

- Data
- Noise

$n = 10$  runs

## Conclusions

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