Neural Distance Embeddings for Biological Sequences

Gabriele Corso, Rex Ying, Michal Pándy, Petar Veličković, Jure Leskovec, Pietro Liò
NeurIPS 2021

CS886 Winter 2022 - Deep Learning for Biotechnology
Presenter: **Gustavo Sutter (***gsutterp@uwaterloo.ca)*

Distance between biological sequences

- Over the course of evolution biological sequences mutate and became different from each other
- Biologists have developed methods to **estimate their evolutionary distance based on their edit distance** D(s1, s2)
- Several tasks can be performed using those distances
  - Hierarchical Clustering
  - Multiple sequence alignment (MSA)
  - Steiner string

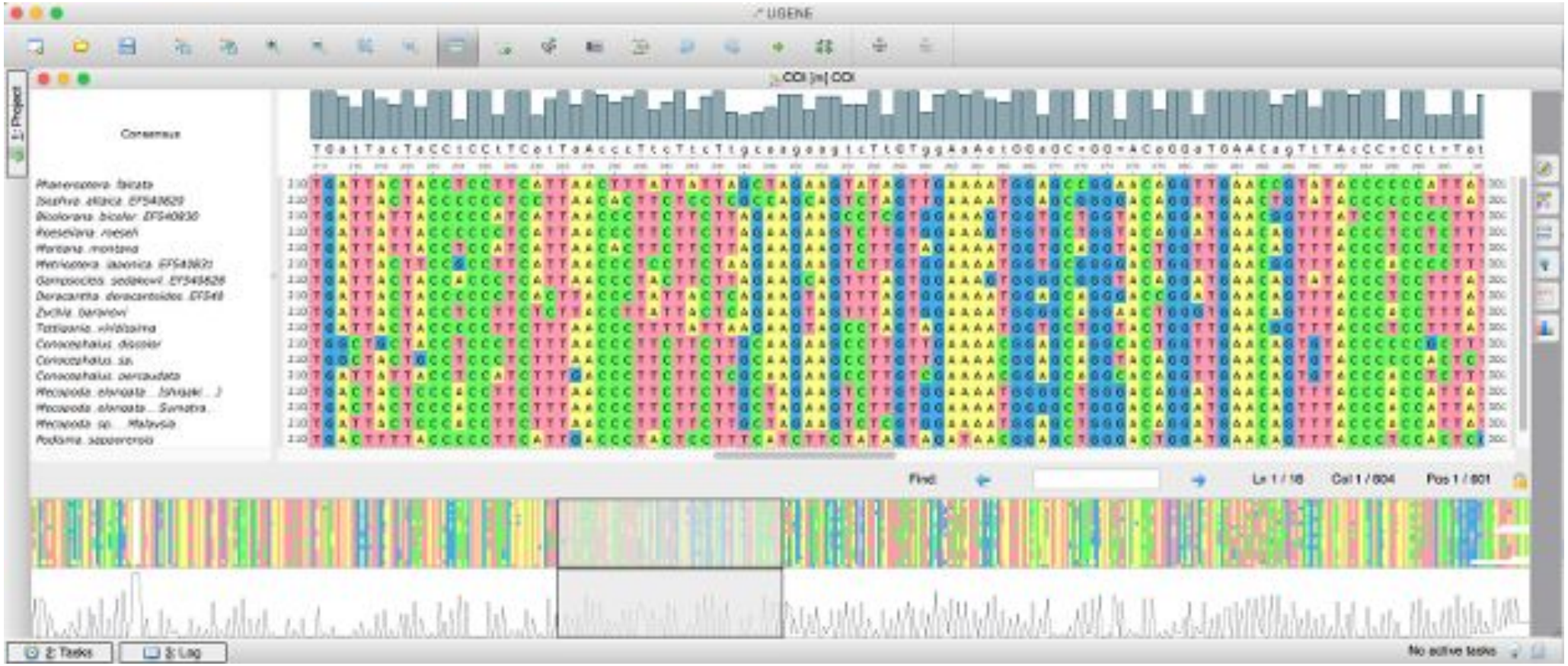
Hierarchical Clustering (HC)

- The goal is to discover the **intrinsic hierarchical structure given by evolutionary history**
- The algorithm uses **agglomerative clustering** to construct a phylogenetic tree
- In order to the method requires a distance matrix indicating the **distance between every pair of sequences** on the dataset

Multiple sequence alignment (MSA)

- Align three or more sequences
- Used for the identification of active and binding site as well as conserved protein structures → homology
- NP-Complete problem
- Clustal is the most popular MSA heuristic
  - Phase 1: Construct phylogenetic tree using HC
  - Phase 2: Progressive alignment

Multiple sequence alignment (MSA)



UNIVERSITY OF WATERLOO

Steiner string

- The Steiner string is the string the minimizes the sum of distances (consensus error) to a set of strings

$$s^* = argmin_s \sum_{s_i \in S} ED(s, s_i)$$

- Hard to do in edit space but can trivial in other spaces (e.g., Euclidean)

Issues with finding the edit distance

- Classical algorithms find the edit distance in **alignment-based** manner, using dynamic programming
  - Needleman-Wunsch algorithm
- However those methods are **quadratic** with respect to the length of the input sequence
- To address this problem **alignment-free** methods were developed

k-mer

- The idea behind *k*-mer is to count the number of occurrences of all subsequences of length *k*
- Once with the number of occurrences they are put into a vector that is used to represent the sequence
- FFP is a method based on the Jensen-Shannon divergence between *k*-mers

k-mer: Example

A U A U C G U A A U C G

2-mer:

AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

k-mer: Example

A U A U C G U A A U C G

2-mer:

AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

k-mer: Example

A U A U C G U A A U C G

2-mer:

AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

k-mer: Example

A U A U C G U A A U C G

2-mer:

AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0

k-mer: Example

A U A U C G U A A U C G

2-mer:

AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
0	0	0	2	0	0	0	0	0	0	0	0	1	1	0	0

k-mer: Example

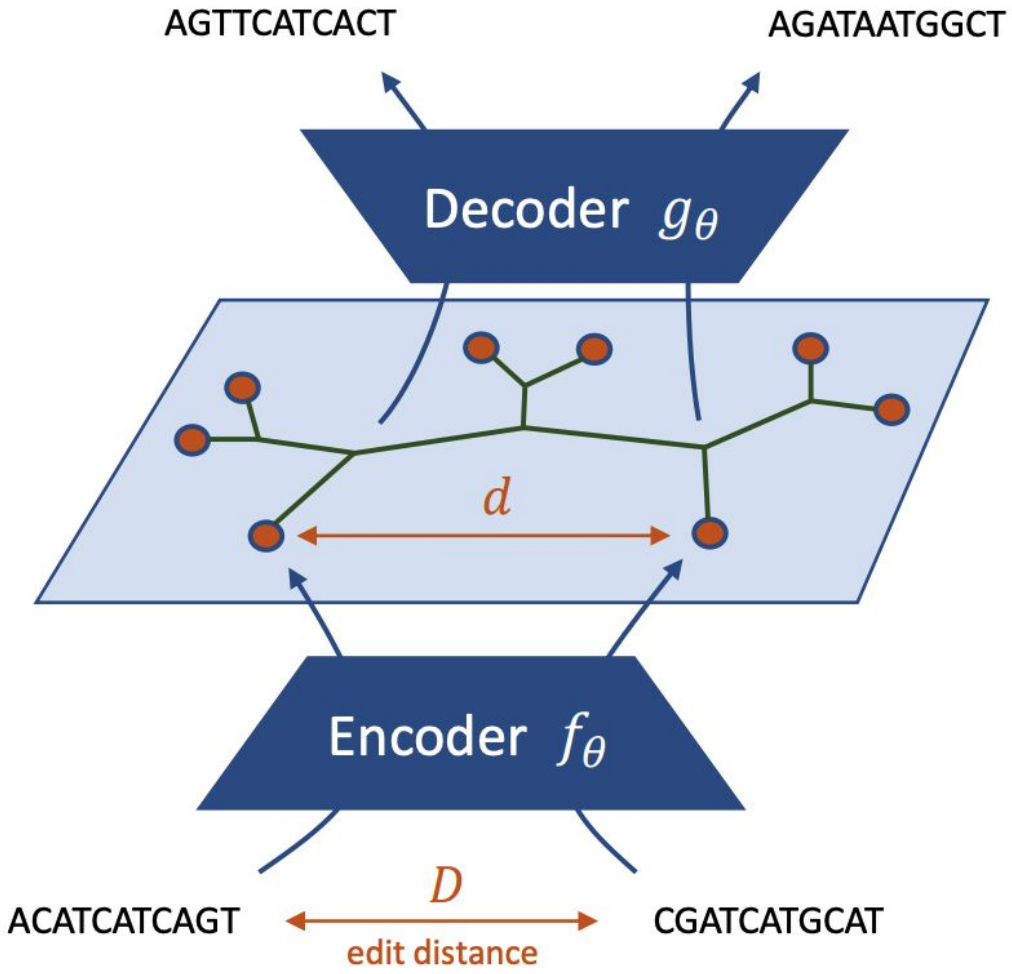
A U A U C G U A A U C G

2-mer:

AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
1	0	0	3	0	0	2	0	0	0	0	1	2	2	0	0

Neural networks

- Neural nets are function approximators, so we can use them to **learn the edit distance**
- Map sequences in a continuous space where the distance between the embedded points is correlated with the one between the sequences

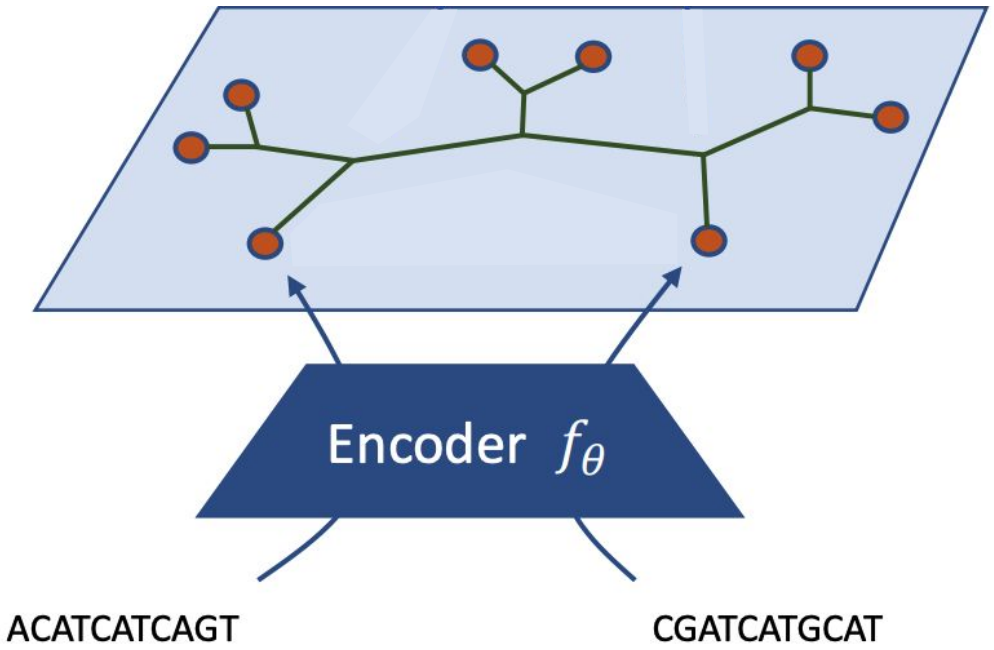


Neural networks: embedding geometry

- The embedding geometry is defined by the distance function of the embedding space
  - Euclidean
  - Manhattan
  - Cosine
  - Square Euclidean
  - Hyperbolic

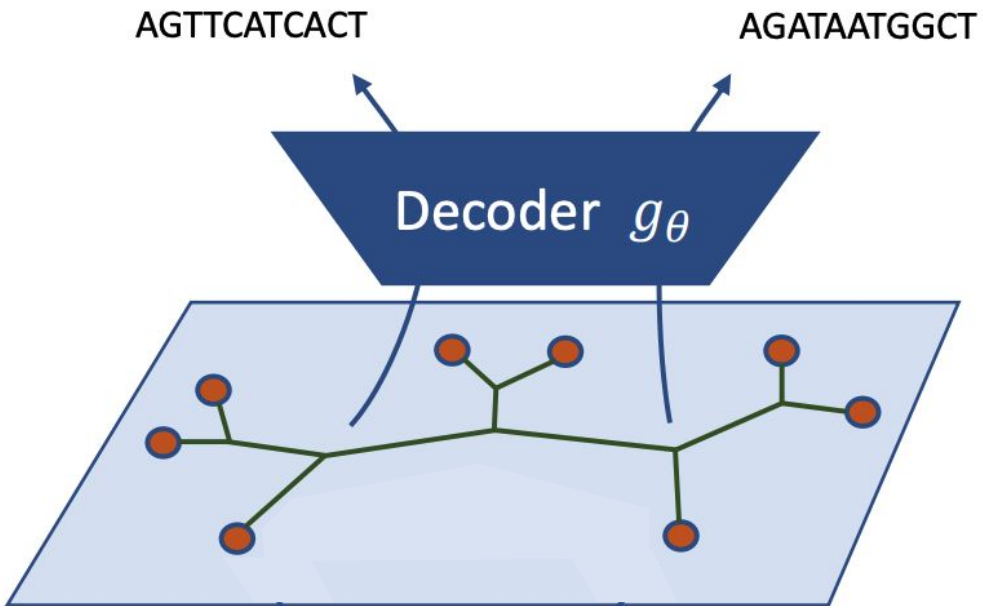
Neural networks: encoder model

- The encoder model maps sequences to points in the embedding space
  - Linear layer
  - MLP
  - CNN
  - GRU
  - Transformer (global or local attention)



Neural networks: decoder model

- For some tasks it is useful to decode points from the embedding space back to sequences



Neural networks: loss function

- The simplest loss is the MSE between the sequence's distance and its approximation as the distance between the embedding

$$L(\theta, S) = \sum_{s_1, s_2 \in S} \left(D(s_1, s_2) - \alpha\, d(f_\theta(s_1), f_\theta(s_2))\right)^2$$

Neural networks

Table 1: Summary of the previous and the proposed `NeuroSEED` approaches. EDA stands for edit distance approximation and CSR for closest string retrievals. For our experiments, in the columns geometry and encoder we report those that performed best among the ones tested.

Method	Geometry	Encoder	Decoder	Loss	Tasks
Zheng *et al.* [11]	Jaccard	CNN	✗	MSE	EDA
Chen *et al.* [24]	Cosine	CSM	✗	MSE	EDA
Zhang *et al.* [25]	Euclidean	GRU	✗	MAE + triplet	EDA & CSR
Dai *et al.* [26]	Euclidean	CNN	✗	MAE + triplet	EDA & CSR
Gomez *et al.* [27]	Square	CNN	✗	MSE	EDA & CSR
Section 5	Hyperbolic	CNN & transformer	✗	MSE	EDA
Section 6	Hyperbolic	CNN & transformer	✗	MSE	HC & MSA
Section 7.1	Hyperbolic	Linear	✗	Relaxed Dasgupta	HC
Section 7.2	Cosine	Linear	✓	MSE + reconstr.	MSA
Appendix F	Hyperbolic	CNN & transformer	✗	MSE & triplet	CSR

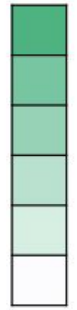
Edit distance approximation

- Three datasets of 16S rRNA
  - RT988 (6.7k sequences with up to 465 bp)
  - Quiita (6M sequences with up to 152 bp)
  - Greengenes (1M sequences with up to 2368 bp)
- Methods
  - k-mer
  - FFP
  - Various neural network approaches

Edit distance approximation

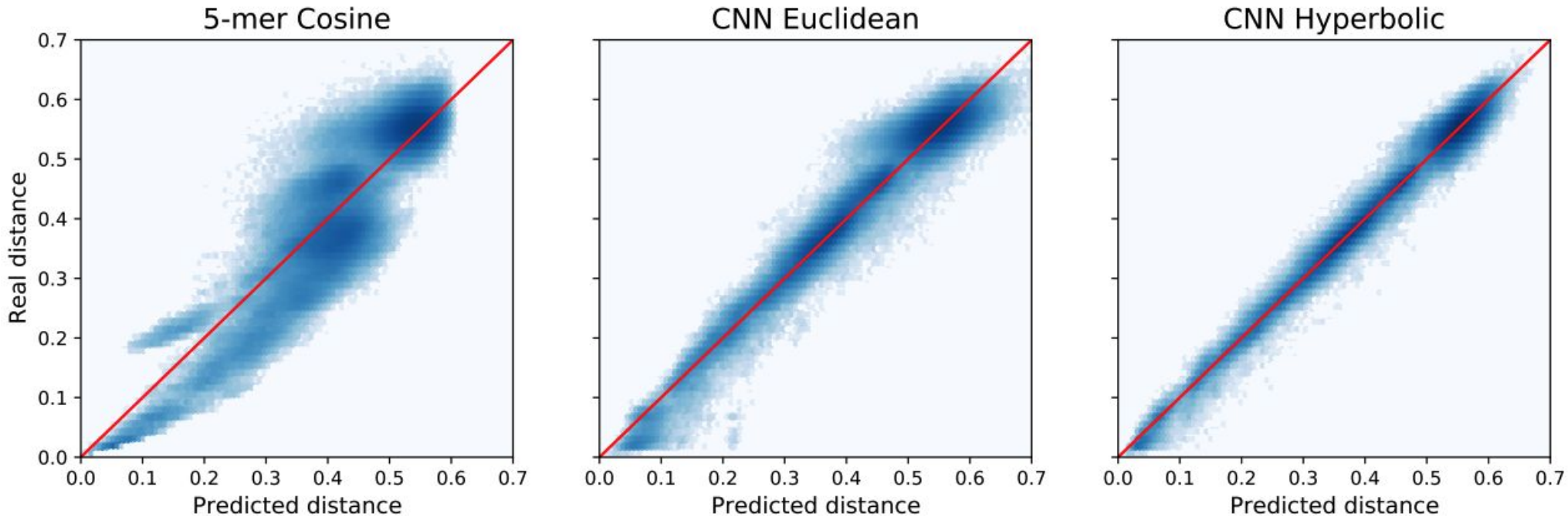
	RT988		Qiita		Greengenes		
Model	**Baseline**	**Hyperbolic**	**Baseline**	**Hyperbolic**	**Baseline**	**Hyperbolic**	**Training/Inference**
NW alignment	-	-	-	-	-	-	- / 17.5h
4-mer	1.79	-	6.01	-	5.93	-	7s / 7s
5-mer	1.41	-	5.03	-	3.60	-	29s / 29s
6-mer	1.47	-	5.72	-	3.15	-	118s / 118s
FFP 8	12.03	-	20.42	-	10.26	-	360s / 360s
FFP 9	11.86	-	17.53	-	8.63	-	679s / 679s
FFP 10	10.80	-	16.16	-	14.13	-	1274s / 1274s
Linear	21.36±7.07	0.51±0.01	4.39±0.09	2.50±0.01	1155.74±18.34	2.70±0.01	1.1h / 3s
MLP	1.10±0.05	0.59±0.20	4.36±0.19	1.85±0.02	4.38±0.13	2.53±0.03	0.9h / 3s
CNN	0.58±0.05	0.59±0.01	2.68±0.05	**1.56±0.01**	1.37±0.04	**1.00±0.01**	2.1h / 6s
GRU	1.10±0.11	2.56±3.33	3.30±0.06	2.60±0.16	1.61±0.02	1.18±0.16	7.4h / 65s
Global T.	0.52±0.01	**0.46±0.01**	2.10±0.05	1.83±0.03	2.09±0.03	1.91±0.07	2.2h / 3s
Local T.	0.57±0.00	**0.45±0.01**	2.42±0.02	1.86±0.02	1.85±0.04	1.89±0.05	2.0h / 3s

Best



Worst

Edit distance approximation



UNIVERSITY OF WATERLOO

Edit distance approximation: data-dependency

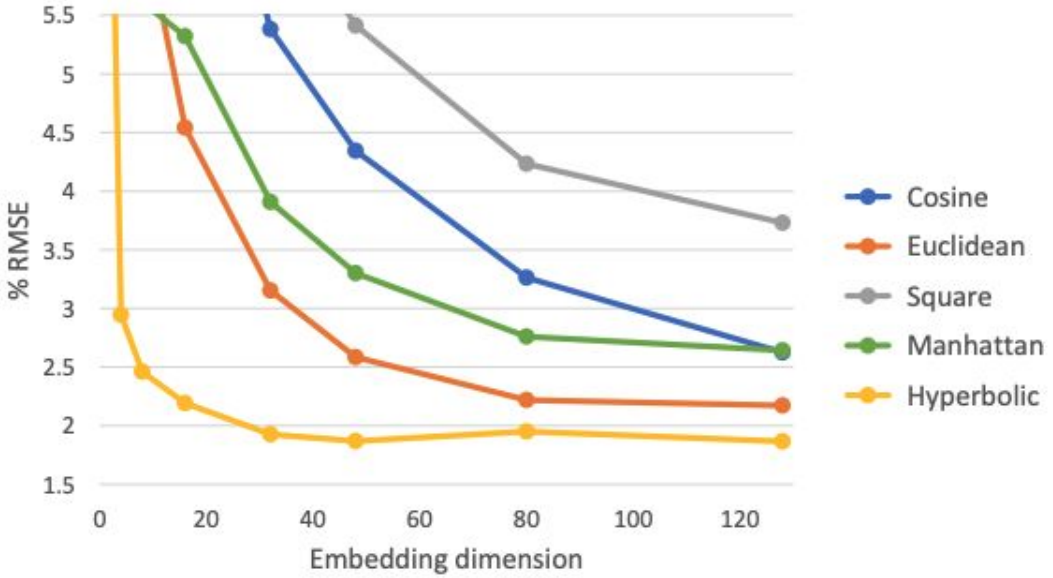
- Data-dependent models learn better representations because they can focus on the important parts of the sequences and learn a lower dimension manifold
- To demonstrate this hypothesis the authors investigate the performance of the models when using random synthetic data for training

Edit distance approximation: data-dependency

	RT988					Qiita					Synthetic					Greengenes				
Model	**Cosine**	**Euclidean**	**Square**	**Manhattan**	**Hyperbolic**	**Cosine**	**Euclidean**	**Square**	**Manhattan**	**Hyperbolic**	**Cosine**	**Euclidean**	**Square**	**Manhattan**	**Hyperbolic**	**Cosine**	**Euclidean**	**Square**	**Manhattan**	**Hyperbolic**
2-mer	7.782	4.927	8.000	5.036	4.859	21.222	11.752	30.453	11.639	10.481	10.49	7.11	10.53	7.28	7.11	16.172	7.983	14.753	7.931	5.084
3-mer	3.392	3.351	3.520	2.987	3.308	12.352	7.962	32.219	7.439	6.657	5.71	6.02	5.81	6.01	5.99	11.210	5.583	10.994	5.352	5.133
4-mer	1.790	3.314	1.899	2.318	3.294	6.006	7.015	34.098	5.636	6.728	**3.74**	6.24	3.87	5.92	6.23	5.931	3.874	5.981	3.611	5.164
5-mer	1.409	3.449	1.422	1.801	3.470	5.027	7.638	34.559	5.391	7.600	3.92	6.75	3.97	5.72	6.75	3.600	3.427	3.339	3.107	5.182
6-mer	1.471	3.710	1.450	1.686	3.730	5.723	8.383	34.616	5.844	8.275	4.71	7.26	4.72	5.37	7.31	3.152	3.478	2.828	2.905	5.192
Linear	0.62±0.03	21.3±7.0	27.2±10.8	-	0.51±0.01	3.38±0.06	4.39±0.09	5.83±0.21	3.82±0.09	2.50±0.01	4.77±0.04	33.9±35.1	5.25±0.03	-	6.50±0.60	3.60±0.05	1155±18.3	2670±3209	14133±680	2.70±0.01
MLP	1.57±0.16	1.10±0.05	6.78±2.50	1.01±0.04	0.59±0.20	4.98±0.11	4.36±0.19	8.52±0.78	4.92±0.10	1.85±0.02	9.79±0.08	9.40±0.05	7.74±0.05	9.82±0.06	10.71±0.18	4.60±0.08	4.38±0.13	8.73±0.77	3.97±0.06	2.53±0.03
CNN	0.69±0.03	0.58±0.05	2.95±1.09	0.98±0.06	0.59±0.01	2.54±0.04	2.68±0.05	5.03±0.85	4.06±0.21	**1.56±0.01**	4.18±0.25	4.93±0.04	4.93±0.03	5.48±0.06	4.60±0.15	1.83±0.05	1.37±0.04	2.23±0.03	1.58±0.03	**1.00±0.01**
GRU	14.9±4.56	1.10±0.11	1.96±0.47	1.13±0.15	2.56±3.33	-	3.30±0.06	5.52±0.15	3.74±0.01	2.60±0.16	6.30±4.93	5.11±0.10	5.60±4.33	5.68±0.22	8.54±0.84	24.69±0.00	1.61±0.02	24.69±0.00	4.90±0.69	1.18±0.16
Global T.	0.49±0.01	0.52±0.01	0.88±0.02	**0.44±0.01**	**0.46±0.01**	2.61±0.01	2.10±0.05	3.71±0.04	2.57±0.11	1.83±0.03	4.51±0.01	4.74±0.02	5.23±0.03	4.67±0.04	4.75±0.04	2.16±0.04	2.09±0.03	2.83±0.04	1.73±0.03	1.91±0.07
Local T.	0.51±0.03	0.57±0.00	0.58±0.02	0.48±0.01	**0.45±0.01**	2.67±0.04	2.42±0.02	3.72±0.06	2.46±0.02	1.86±0.02	4.45±0.03	4.86±0.03	5.05±0.03	4.87±0.02	4.49±0.03	2.12±0.02	1.85±0.04	2.37±0.05	1.72±0.06	1.89±0.05

Edit distance approximation: hyperbolic space

- Biological dataset have implicit hierarchical structure that is reflected by the hyperbolic space
- The hyperbolic space provide significantly more efficient embeddings



$$d(\mathbf{p},\mathbf{q}) = \operatorname{arcosh}\left(1 + 2\frac{\|\mathbf{p}-\mathbf{q}\|^2}{(1-\|\mathbf{p}\|^2)(1-\|\mathbf{q}\|^2)}\right)$$

Edit distance approximation: computational complexity

- Assuming constant embedding size and a model linear with respect to the sequence length the complexity for computing the pairwise distance matrix is $O(N(M+N))$
  - Previously it was $O(N^2M^2/\log M)$

Hierarchical Clustering

- Once with a matrix of approximated pairwise distances it is possible to perform hierarchical clustering
- The authors trained the models on the Quiita dataset and used a different holdout dataset to evaluate the algorithms performance

Hierarchical Clustering

Baselines	
Single L.	0.628
Complete L.	0.479
Average L.	**0.000**

Model	Cosine	Euclidean	Square	Manhattan	Hyperbolic
4-mer	0.261	0.260	0.242	0.191	0.299
Linear	0.062±0.007	0.172±0.036	0.153±0.037	0.177±0.026	0.028±0.005
MLP	0.169±0.054	0.095±0.021	0.289±0.094	0.178±0.029	0.035±0.004
CNN	0.028±0.003	0.030±0.004	0.067±0.022	0.081±0.047	**-0.004±0.015**
GRU	-	0.042±0.006	0.068±0.010	0.069±0.015	0.066±0.043
Global T.	0.032±0.014	**0.003±0.008**	0.038±0.005	**0.002±0.003**	**0.000±0.006**
Local T.	0.035±0.003	0.022±0.008	0.034±0.005	0.022±0.003	**0.000±0.007**

Figure 5: Average Linkage % increase in Dasgupta's cost of NeuroSEED models compared to the performance of clustering on the ground truth distances, ubiquitously used in bioinformatics. Average Linkage was the best performing clustering heuristic across all models.

Multiple sequence alignment (MSA)

- As mentioned previously, the first step in MSA relies on hierarchical clustering
- Building a phylogenetic tree using the distance between embeddings and passing it to the Clustal alignment algorithm allows faster computation
- The results exhibit high variance
  - Authors suggest using ensemble of models to overcome this issue

Multiple sequence alignment (MSA)

Model	Cosine	Euclidean	Hyperbolic
Linear	60.6±35.1	111.3±3.6	57.5±22.0
MLP	72.3±11.8	53.6±3.1	-11.7±18.9
CNN	31.0±16.2	4.7±9.7	**-16.3±16.1**
Global T.	39.4±74.3	1.9±3.8	31.1±21.8
Local T.	31.9±30.5	8.6±14.1	**-20.1±7.3**

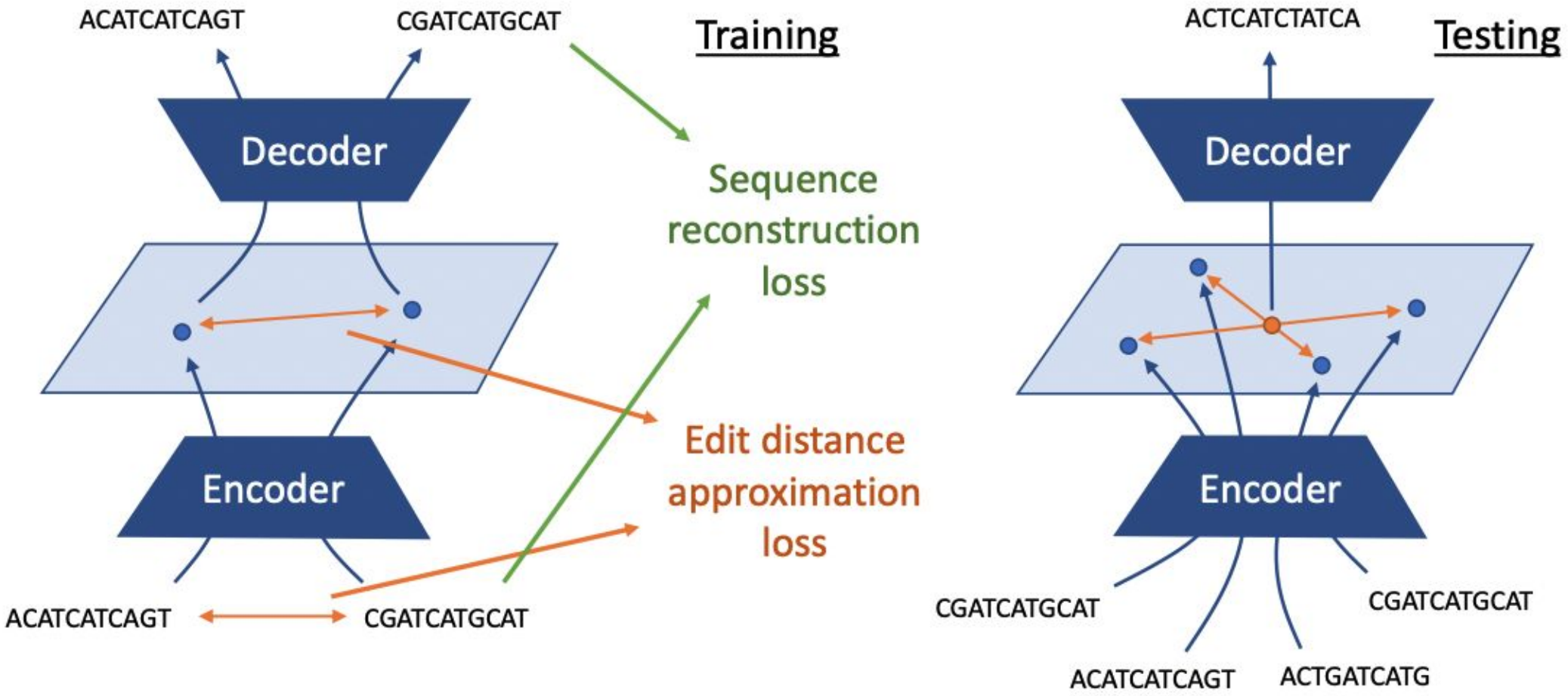
Figure 6: Percentage improvement (average of 3 runs) in the alignment cost (the lower the better) returned by Clustal when using the heuristics to generate the tree as opposed to its default setting using NJ on real distances.

Steiner string

- The idea is to **compute the Steiner string on the embedding** space and decode it back to a sequence
- The model is trained to approximate the distance in the embedding space and to reconstruct the sequences after decoding

$$L(\theta, \theta') = \underbrace{(1-\alpha)\, L_{\mathrm{ED}}(\theta)}_{\text{edit distance}} + \underbrace{\alpha\, L_{\mathrm{R}}(\theta, \theta')}_{\text{reconstruction}}$$

Steiner string



UNIVERSITY OF WATERLOO

Steiner string

Baselines	
Random	75.98
Centre	62.52
Greedy-1	**59.43**
Greedy-2	**59.41**

Model	Cosine	Euclidean	Square	Hyperbolic
Linear	**59.41±0.11**	59.96±0.27	60.53±0.49	60.89±0.82
MLP	60.80±0.35	60.00±0.18	59.81±0.22	59.86±0.12
CNN	60.96±0.48	60.20±0.26	60.76±1.09	60.48±0.52

Figure 9: Average consensus error for the baselines (left) and NeuroSEED models (right).

Conclusion

- In this work the authors propose a framework that applies advances in representation learning to embed biological sequences
- They show the strong advantage provided by the hyperbolic space
- Demonstrate the capacity of their method for HC, MSA and Steiner string discovery, fundamental problems in genomics

Thank you!