MIK: Modified Isolation Kernel for Biological Sequence Visualization, Classification, and Clustering

Presented By

SARRWAN ALI

Georgia State University

November 13, 2024

Table of Contents

Introduction

- Predicting the impact of amino acid changes on protein function is essential for applications such as disease variant classification and protein engineering
- Computing pairwise sequence similarity becomes more important for protein function prediction, data visualization, and pattern recognition
- The t-distributed stochastic neighbor embedding (t-SNE) :
    - It is a method for interpreting high dimensional (HD) data by mapping each point to a low dimensional (LD) space (usually two-dimensional)
    - Used for better visualization
    - Dimensionality reduction

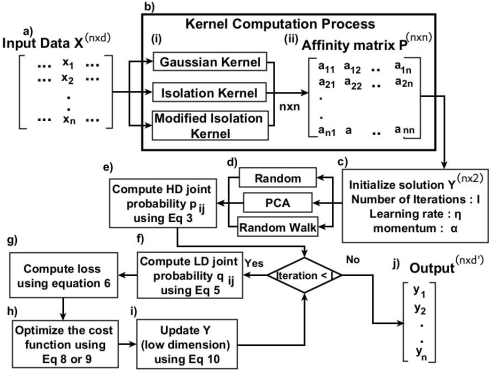
Motivation

- High dimensionality data in biological sequences
- Better low dimensional Visualization
- Improve performance and reduce computational cost
- The vast global spread of pandemics like COVID-19, pushing viral sequence analysis into the “Big Data” realm

Our (High Level) Idea

- We propose the Modified Isolation Kernel (MIK), as an alternative to the Gaussian and Isolation kernel
- It is intended to address the existing shortcomings in preserving local and global structures, and handling noisy data and outliers
- MIK is evaluated using a variety of initialization techniques
  - Random initialization
  - PCA-based initialization
  - Random walk-based initialization
- The random walk-based initialization for such biological data is not been explored in the literature

Workflow


a)
Input Data $X^{(nxd)}$
... x_1 ...
... x_2 ...
.
.
... x_n ...
b)
Kernel Computation Process
(i)
Gaussian Kernel
Isolation Kernel
Modified Isolation Kernel
nxn
(ii) Affinity matrix $P^{(nxn)}$
a_{11} a_{12} .. a_{1n}
a_{21} a_{22} .. a_{2n}
a_{n1} a .. a_{nn}
c)
Initialize solution $Y^{(nx2)}$
Number of Iterations : I
Learning rate : η
momentum : α
d)
Random
PCA
Random Walk
e)
Compute HD joint probability p_{ij} using Eq 3
f)
Compute LD joint probability q_{ij} using Eq 5
Yes
Iteration < I
No
g)
Compute loss using equation 6
h)
Optimize the cost function using Eq 8 or 9
i)
Update Y (low dimension) using Eq 10
j) $Output^{(nxd')}$
y_1
y_2
.
.
y_n

Problem Formulation

- Given dataset $X = \{x_1, x_2, ..., x_n\}$ in $\mathbb{R}^d$
- Assume a dataset $Y = \{y_1, y_2, ..., y_n\}$
- Objective: Map $X \in \mathbb{R}^d$ to $Y \in \mathbb{R}^{d'}$, such that $d' < d$
- $d' = 2$ or 3
- The similarity between points is preserved as much as possible
- Goal: Map points from X to Y such that the probability distribution between P_{ij} and Q_{ij} are as close as possible
- The similarity between a pair of points x_i, x_j in the higher dimensional space is represented by a probability P_{ij}
- The similarly for low dimensional space points y_i, y_j is represented by Q_{ij}

Gaussian Kernel

$$K(x_i, x_j) = exp(\frac{-||x_i - x_j||^2}{2\sigma_i^2})$$

Isolation Kernel

It measures the isolation of a data point from its neighbors

1. Pairwise Squared Distances:

$$D_{ij} = \|x_i - x_j\|^2$$

Isolation Kernel (Contd.)

2 Perplexity and Effective Neighborhood Size:

$$H(P_i) = \log\left(\sum_j P_{ij}\right) + \beta\frac{\sum_j D_{ij}P_{ij}}{\sum_j P_{ij}}$$

- $H(P_i)$ represents the entropy of P_i for the i-th point
- $P_{ij} = \exp(-\beta D_{ij})$ represents the similarity between points x_i and x_j

Isolation Kernel (Contd.)

3 Scaling Parameter (Beta) Adjustment:

$$\beta = \frac{\text{Perplexity}}{\text{distance scaling} \cdot (\max(D_i) + \epsilon)}$$

This beta adjustment is repeated until $H(P_i) \approx \log(\text{Perplexity})$

Isolation Kernel (Contd.)

4. Kernel Matrix Construction:

$$P_{ij} = \frac{P_{ij}}{\sum_j P_{ij}}$$

   where each row in P sums to 1, providing a probabilistic interpretation of the kernel

MIK

The Modified Isolation Kernel incorporates two additional components: a *distance scaling* factor and *weights* for each point

- The *distance scaling* factor is computed based on the average pairwise distance between all points in the dataset $X = \{x_1, x_2, \ldots, x_n\}$:

  $$D_{ij} = \|x_i - x_j\|$$

  where D_{ij} is the Euclidean distance between points x_i and x_j. The distance scaling factor s is then calculated as:

  $$s = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} D_{ij}$$

  This scaling factor is used to normalize the distances, ensuring that the neighborhood sizes are comparable across different datasets

MIK (Contd.)

- Weights Adjustment: After normalizing P_i for each point i, weights w_i are applied to modulate the values in P:

  $$P_{ij} = \frac{P_{ij}}{\sum_j P_{ij}} \cdot w_i$$

  where:
  - w_i is a weight applied to row i to adjust its influence on the kernel

Weights Computation Using DBSCAN

- The weights are computed based on the density of points around each data point, using the **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) algorithm.
- The DBSCAN algorithm assigns a weight (or label) to each point according to its neighborhood density. The parameters for DBSCAN include:
  - Epsilon ϵ: the maximum distance between two points to be considered neighbors
  - min samples: the minimum number of points required in a neighborhood for a point to be considered a core point

Weights Computation Using DBSCAN (Contd.)

- Given the dimensionality of the data d, the minimum samples parameter min samples is set to d + 1
- This ensures that the density estimation accounts for the dimensionality of the dataset, with ϵ chosen based on dataset-specific properties
- The weights w_i for each point x_i are defined as the labels assigned by DBSCAN:

  $$w_i = \text{DBSCAN}(x_i)$$

  where points labeled as noise by DBSCAN receive a weight of -1.
- These weights can then be incorporated into the Isolation Kernel to emphasize regions with higher or lower point densities

DBSCAN-based Weights

$$w_i = \begin{cases} \text{cluster label of } x_i & \text{if } x_i \text{ is in a dense region} \\ -1 & \text{if } x_i \text{ is labeled as noise} \end{cases}$$

- This approach helps to adjust the kernel by accounting for both the average distances and the density-based clustering within the dataset

Kernel Comparison

- **Isolation Kernel:**

$$P_{ij} = \frac{\exp(-\beta D_{ij})}{\sum_j \exp(-\beta D_{ij})}$$

- **Modified Isolation Kernel with Distance Scaling and Weights:**

$$P_{ij} = \frac{\exp(-\beta D_{ij}^{\text{scaled}})}{\sum_j \exp(-\beta D_{ij}^{\text{scaled}})} \cdot w_i$$

Where:

  - β is adjusted iteratively to match the perplexity
  - $D_{ij}^{\text{scaled}} = D_{ij} \cdot$ distance-scale adjusts the distance, and w_i applies optional point-specific weights

t-SNE Evaluation Metrics

- Neighborhood Agreement (NA):

$$NA = 1 - \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j \neq i} \left| \frac{d_{ij}^{H} - d_{ij}^{L}}{d_{ij}^{H} + d_{ij}^{L}} \right| \tag{1}$$

- Trustworthiness (TW) :

$$TW = 1 - \frac{2}{N \cdot k \cdot (2N - 3k - 1)} \sum_{i=1}^{N} \sum_{j \in R_k} (R_{ij} - R_{ij}^{L}) \tag{2}$$

Classification Evaluation Metrics

- We use Support Vector Machine (SVM), Naive Bayes (NB), Multi-Layer Perceptron (MLP), K-Nearest Neighbour (KNN), Random Forest (RF), Logistic Regression (LR), and Decision Tree (DT) classifiers
- We use average accuracy, precision, recall, weighted, and ROC area under the curve (AUC) as evaluation metrics for measuring the goodness of classification algorithms

Clustering Evaluation Metrics

- **Silhouette Coefficient** [1]
- **Calinski-Harabasz Index** [2]
- **Davies-Bouldin Index** [3]

Dataset

Dataset	| Seq. |	| Classes |	Sequence Length			Detail
			Max	Min	Mean	
Protein Subcellular [4]	5959	11	3678	9	326.27	The unaligned protein sequences having information about subcellular locations.
GISAID [5]	7000	22	1274	1274	1274.00	The aligned spike sequences of the SARS-CoV-2 virus having the information about the Lineage of each sequence.
Nucleotide [6]	4380	7	18921	5	1263.59	Unaligned nucleotide sequences to classify gene family to which humans belong

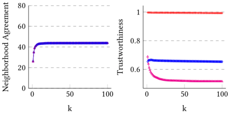
Embedding Methods

- **Spike2Vec** [7]
- **Spaced** *k*-**mer** [8]
- **PWM2Vec** [9]

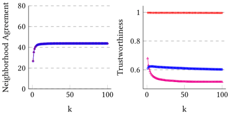
Initialization Methods

- Random Initialization
- PCA-based Initialization
- Random Walk-based Initialization

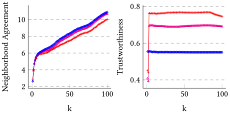
Results - Nucleotide Dataset



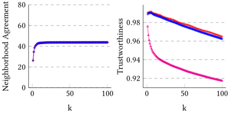
(a) Spike2Vec (Random Init.)



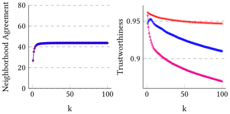
(b) Spaced *k*-mers (Random Init.)



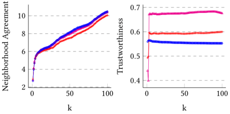
(c) PWM2Vec (Random Init.)



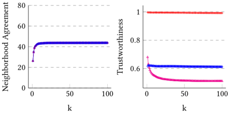
(d) Spike2Vec (PCA Init.)



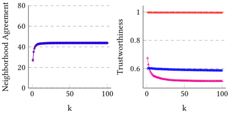
(e) Spaced *k*-mers (PCA Init.)



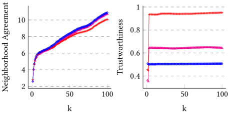
(f) PWM2Vec (PCA Init.)



(g) Spike2Vec (Random Walk Init.)



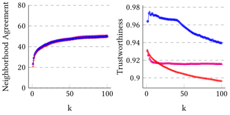
(h) Spaced *k*-mers (Random Walk Init.)



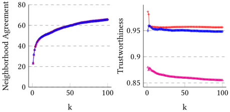
(i) PWM2Vec (Random Walk Init.)

— Gaussian
— Isolation
— Weighted Isolation

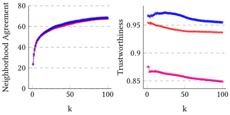
Results - GISAID Dataset



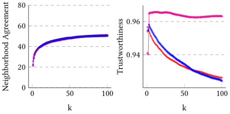
(a) Spike2Vec (Rand.)



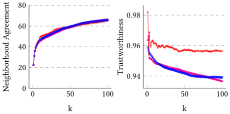
(b) Spaced *k*-mers (Rand.)



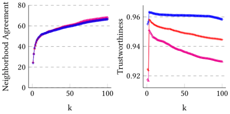
(c) PWM2Vec (Rand.)



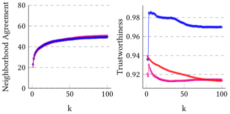
(d) Spike2Vec (PCA)



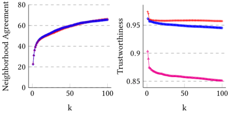
(e) Spaced *k*-mers (PCA)



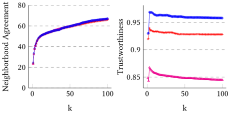
(f) PWM2Vec (PCA)



(g) Spike2Vec (Walk)



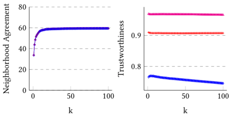
(h) Spaced *k*-mers (Walk)



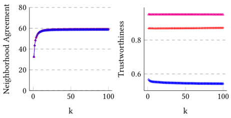
(i) PWM2Vec (Walk)

— Gaussian
— Isolation
— Weighted Isolation

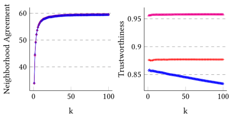
Results - Protein Subcellular Dataset



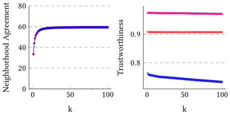
(a) Spike2Vec (Random Init.)



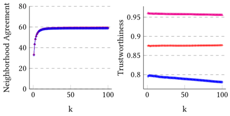
(b) Spaced *k*-mers (Random Init.)



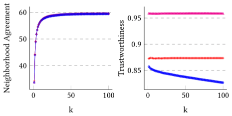
(c) PWM2Vec (Random Init.)



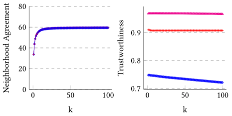
(d) Spike2Vec (PCA Init.)



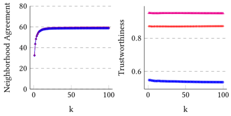
(e) Spaced *k*-mers (PCA Init.)



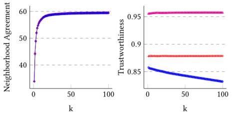
(f) PWM2Vec (PCA Init.)



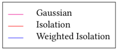
(g) Spike2Vec (Random Walk Init.)



(h) Spaced *k*-mers (Random Walk Init.)



(i) PWM2Vec (Random Walk Init.)



Classification Results - Protein Subcellular Dataset

Kernel	Embeddings	Algo.	Acc. ↑	Prec. ↑	Recall ↑	F1 (Weig.) ↑	F1 (Macro) ↑	ROC AUC ↑	Train Time (sec.) ↓
Gaussian	PWM2Vec	SVM	0.5207	0.5190	0.5207	0.5110	0.3947	0.6630	9.7228
		NB	0.3876	0.4438	0.3876	0.3965	0.3206	0.6376	0.0554
		MLP	0.4640	0.4424	0.4640	0.4487	0.2990	0.6196	9.8790
		KNN	0.5634	0.5671	0.5634	0.5550	0.4567	0.6950	0.1219
		RF	0.5252	0.6064	0.5252	0.4835	0.3176	0.6215	6.8668
		LR	0.5085	0.4830	0.5085	0.4654	0.2709	0.6106	0.4873
		DT	0.3668	0.3702	0.3668	0.3677	0.2597	0.5978	0.6648
Isolation	Spike2Vec	SVM	0.6493	0.6550	0.6493	0.6487	0.4977	0.7326	9.7372
		NB	0.2701	0.3615	0.2701	0.2285	0.2426	0.6714	0.0455
		MLP	0.7796	0.7727	0.7796	0.7752	0.6105	0.7917	11.8214
		KNN	0.8546	0.8595	0.8546	0.8522	0.7464	0.8622	0.1707
		RF	0.8345	0.8538	0.8345	0.8069	0.5968	0.7712	5.9505
		LR	0.2450	0.0600	0.2450	0.0964	0.0358	0.5000	0.2048
		DT	0.6913	0.6973	0.6913	0.6919	0.5203	0.7430	0.6885
Modified Isolation (Ours)	Spaced *k*-mers	SVM	0.8697	0.8767	0.8697	0.8712	0.7327	0.8663	1.1591
		NB	0.4010	0.5894	0.4010	0.3824	0.3661	0.7481	0.0364
		MLP	0.8937	**0.8971**	0.8937	**0.8949**	**0.7847**	**0.8907**	10.0736
		KNN	0.6119	0.6231	0.6119	0.5914	0.4010	0.6817	0.1299
		RF	**0.8982**	0.8947	**0.8982**	0.8867	0.7125	0.8443	6.3723
		LR	0.2204	0.2584	0.2204	0.0811	0.0337	0.5003	0.2525
		DT	0.7265	0.7377	0.7265	0.7301	0.5701	0.7793	1.0005

Classification Results - GISAID Dataset

Kernel	Embeddings	Algo.	Acc. ↑	Prec. ↑	Recall ↑	F1 (Weig.) ↑	F1 (Macro) ↑	ROC AUC ↑	Train Time (sec.) ↓
Gaussian	Spike2Vec	SVM	0.7397	0.6998	0.7397	0.7009	0.3060	0.6541	3.4177
		NB	0.1564	0.6631	0.1564	0.2226	0.1312	0.5745	**0.0866**
		MLP	0.7696	0.7308	0.7696	0.7388	0.4025	0.7111	8.2361
		KNN	0.7678	0.7725	0.7678	0.7634	0.5223	0.7560	0.1034
		RF	**0.7872**	0.7809	**0.7872**	**0.7762**	0.5342	**0.7637**	2.8961
		LR	0.7174	0.6514	0.7174	0.6686	0.2447	0.6226	1.2611
		DT	0.7756	0.7702	0.7756	0.7668	0.5106	0.7550	0.2956
Isolation	Spaced *k*-mers	SVM	0.4972	0.3674	0.4972	0.3814	0.1007	0.5331	11.9771
		NB	0.0276	0.2183	0.0276	0.0271	0.0314	0.5203	0.1053
		MLP	0.6499	0.6187	0.6499	0.6276	0.1884	0.5922	14.1174
		KNN	0.6490	0.6460	0.6490	0.6356	0.2063	0.5892	0.1331
		RF	0.7029	0.6777	0.7029	0.6799	0.3469	0.6488	6.2961
		LR	0.4883	0.2385	0.4883	0.3205	0.0298	0.5000	0.6894
		DT	0.6379	0.6457	0.6379	0.6404	0.2828	0.6329	0.8870
Modified Isolation (Ours)	PWM2Vec	SVM	0.6820	0.6764	0.6820	0.6762	0.3117	0.6446	12.7645
		NB	0.5952	0.6176	0.5952	0.5962	0.2017	0.6008	0.1153
		MLP	0.6741	0.6552	0.6741	0.6584	0.2493	0.6194	13.1291
		KNN	0.6559	0.6493	0.6559	0.6453	0.2222	0.5966	0.1637
		RF	0.7103	0.6955	0.7103	0.6983	0.3612	0.6619	12.8492
		LR	0.4748	0.2255	0.4748	0.3057	0.0293	0.5000	0.6317
		DT	0.6709	0.6752	0.6709	0.6717	0.3168	0.6514	1.6763

Classification Results - Nucleotide Dataset

Kernel	Embeddings	Algo.	Acc. ↑	Prec. ↑	Recall ↑	F1 (Weig.) ↑	F1 (Macro) ↑	ROC AUC ↑	Train Time (sec.) ↓
Gaussian	Spaced *k*-mers	SVM	0.3688	0.6767	0.3688	0.2707	0.2116	0.5447	1.0749
		NB	0.1317	0.7136	0.1317	0.1384	0.1435	0.5411	0.0125
		MLP	0.3857	0.7391	0.3857	0.2926	0.2315	0.5533	3.1893
		KNN	0.2697	0.3876	0.2697	0.2560	0.2308	0.5560	0.0560
		RF	0.4295	0.6904	0.4295	0.3664	0.3188	0.5873	2.0117
		LR	0.3612	0.7508	0.3612	0.2478	0.1825	0.5360	0.1341
		DT	0.4285	0.6625	0.4285	0.3663	0.3198	0.5880	0.2364
Isolation	Spike2Vec	SVM	0.3213	0.3221	0.3213	0.3079	0.2588	0.5728	3.8166
		NB	0.2323	0.4337	0.2323	0.1836	0.2043	0.5560	0.0123
		MLP	0.5275	0.5203	0.5275	0.5192	0.4616	0.6858	7.3494
		KNN	0.5283	0.5340	0.5283	0.5275	0.4976	0.7093	0.0585
		RF	0.7469	0.7539	0.7469	0.7452	0.7345	0.8314	3.4126
		LR	0.3105	0.0965	0.3105	0.1472	0.0677	0.5000	0.0795
		DT	0.6151	0.6161	0.6151	0.6150	0.5866	0.7605	0.3282
Modified Isolation (Ours)	Spaced *k*-mers	SVM	0.5798	0.5762	0.5798	0.5728	0.5358	0.7273	2.6021
		NB	0.2604	0.3666	0.2604	0.2363	0.2463	0.5700	**0.0080**
		MLP	0.6207	0.6208	0.6207	0.6178	0.5734	0.7498	7.1481
		KNN	0.5050	0.5093	0.5050	0.5041	0.4755	0.6954	0.0611
		RF	**0.7481**	**0.7689**	**0.7481**	**0.7455**	**0.7366**	**0.8316**	4.1538
		LR	0.3096	0.1191	0.3096	0.1466	0.0679	0.5001	0.0931
		DT	0.6237	0.6257	0.6237	0.6237	0.5959	0.7644	0.3863

Clustering Results - Protein Subcellular Dataset

Kernel	Embeddings	Algo.	Silhouette ↑	Calinski ↑	Davies ↓
Gaussian	Spike2Vec	K-means	0.101796	360.804549	3.622216
		Agglomerative	0.078076	322.784851	3.585005
		K-Modes	-0.359683	1.646458	1.119053
	Spaced *k*-mers	K-means	0.126160	428.163587	3.500538
		Agglomerative	0.122294	389.211763	3.418970
		K-Modes	-0.309347	4.658780	1.068371
	PWM2Vec	K-means	0.061646	271.209939	3.586904
		Agglomerative	0.019724	229.333382	4.163685
		K-Modes	-0.261891	1.060538	1.035597
Isolation	Spike2Vec	K-means	0.015248	58.296491	2.205289
		Agglomerative	0.100906	66.592869	2.267826
		K-Modes	-0.406924	0.230669	1.709086
	Spaced *k*-mers	K-means	0.894164	49.226179	**0.064535**
		Agglomerative	**0.894826**	54.900365	2.231084
		K-Modes	-0.494872	3.841355	2.039936
	PWM2Vec	K-means	0.427347	57.304634	2.029489
		Agglomerative	0.892029	64.086967	1.472744
		K-Modes	-0.382076	0.146916	1.543540
Modified Isolation (Ours)	Spike2Vec	K-means	0.100461	**875.4463**	1.249022
		Agglomerative	0.258273	856.6747	1.088766
		K-Modes	-0.337161	0.404064	1.154124
	Spaced *k*-mers	K-means	0.298419	766.048709	1.279214
		Agglomerative	0.845733	807.127025	1.059536
		K-Modes	-0.491851	48.420757	2.083230
	PWM2Vec	K-means	0.191296	626.270386	1.425956
		Agglomerative	0.295157	594.130597	1.278188
		K-Modes	-0.358505	0.255160	1.504934

Clustering Results - GISAID Dataset

Kernel	Embeddings	Algo.	Silhouette ↑	Calinski ↑	Davies ↓
Gaussian	Spike2Vec	K-means	0.725709	2360.1012	0.803562
		Agglomerative	0.728701	**2401.6734**	0.897672
		K-Modes	-0.724895	84.479174	1.082510
	Spaced *k*-mers	K-means	0.671649	1758.287930	0.582548
		Agglomerative	0.697097	1750.801213	0.535291
		K-Modes	-0.526676	70.972746	1.769435
	PWM2Vec	K-means	0.691450	1435.243293	0.693299
		Agglomerative	0.660800	1401.089635	0.818952
		K-Modes	-0.520338	94.359037	2.462654
Isolation	Spike2Vec	K-means	0.068962	122.328499	0.912677
		Agglomerative	0.597965	130.992233	2.127489
		K-Modes	-0.594450	0.137340	2.041841
	Spaced *k*-mers	K-means	0.926955	85.446835	1.213278
		Agglomerative	0.935912	90.096434	0.796774
		K-Modes	-0.624410	0.397808	2.425513
	PWM2Vec	K-means	**0.942670**	90.830309	**0.032672**
		Agglomerative	0.940357	91.914837	0.929066
		K-Modes	-0.671301	0.143632	2.922499
Modified Isolation (Ours)	Spike2Vec	K-means	0.062111	191.373940	2.674008
		Agglomerative	0.069198	185.944651	2.621923
		K-Modes	-0.530450	1.363172	1.438185
	Spaced *k*-mers	K-means	0.139867	689.139460	1.760740
		Agglomerative	0.134961	688.940686	1.658960
		K-Modes	-0.461139	0.695404	1.294385
	PWM2Vec	K-means	0.078648	322.821055	2.010462
		Agglomerative	0.058910	312.395730	2.072378
		K-Modes	-0.377294	1.832262	1.199501

Clustering Results - Nucleotide Dataset

Kernel	Embeddings	Algo.	Silhouette ↑	Calinski ↑	Davies ↓
Gaussian	Spike2Vec	K-means	0.845873	100.442214	0.099883
		Agglomerative	0.849057	121.928330	0.097623
		K-Modes	-0.616155	0.680251	12.304512
	Spaced *k*-mers	K-means	0.849962	110.207433	0.097384
		Agglomerative	0.851768	122.335197	**0.096128**
		K-Modes	-0.710487	0.595153	12.299829
	PWM2Vec	K-means	0.906833	1216.9008	0.950717
		Agglomerative	0.907149	1242.8203	1.207521
		K-Modes	-0.783775	19.695747	1.991363
Isolation	Spike2Vec	K-means	0.266999	85.054216	2.373702
		Agglomerative	0.142199	95.157338	2.552974
		K-Modes	-0.398666	0.214001	1.951997
	Spaced *k*-mers	K-means	0.059934	79.041871	2.562520
		Agglomerative	0.045584	86.883693	2.917264
		K-Modes	-0.361089	0.390651	1.595999
	PWM2Vec	K-means	0.493470	63.606950	1.599440
		Agglomerative	**0.928548**	93.383507	0.510479
		K-Modes	-0.535966	6.971815	1.929305
Modified Isolation (Ours)	Spike2Vec	K-means	0.326686	16050.3293	0.781717
		Agglomerative	0.217707	15024.6706	1.035089
		K-Modes	-0.527625	6.309529	1.787152
	Spaced *k*-mers	K-means	0.445578	4740.6237	0.496505
		Agglomerative	0.414100	4529.5243	0.507057
		K-Modes	-0.618379	0.269577	2.518221
	PWM2Vec	K-means	0.906386	85715.7221	0.201059
		Agglomerative	0.906386	**85715.7222**	0.201059
		K-Modes	-0.486018	0.072697	2.195824

Runtime

Kernel	Protein Subcellular	GISAID	Nucleotide
Gaussian	89.12 sec.	6.69 sec.	1.81 sec.
Isolation	135.81 sec.	103.50 sec.	29.80 sec.
MIK	3.34 sec.	2.91 sec.	1.02 sec.
MIK vs. Gaussian % improvement	96.25%	56.50%	43.64%
MIK vs. Isolation % improvement	97.54%	97.18%	96.57%

Recommended Initialization strategies

Dataset	Best Performing	Worst Performing
Protein Subcellular	Random Walk	Random
GISAID	Random Walk	Random
Nucleotide	Random Walk	Random

- Recommendation for initialization method based on the summary of performance on different datasets.

Conclusion

- Modified Isolation Kernel (MIK) as an alternative to the Gaussian kernel, which is built upon the concept of the Isolation Kernel
- MIK uses adaptive density estimation to accurately capture local structures and integrate robustness measures
- Several initialization techniques, including random, PCA, and random walk initializations, are used to assess the proposed approach

Thank You

- P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- T. Caliński and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.
- D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.
- Protein Subcellular Localization, https://www.kaggle.com/datasets/lzyacht/proteinsubcellularlocalization, 2023, [Online; accessed 10-Jan-2023].
- GISAID Website, https://www.gisaid.org/, 2021, [Online; accessed 29-December-2021].

- Human DNA, https://www.kaggle.com/code/nageshsingh/demystify-dna-sequencing-with-machine-learning/data, 2022, [Online; accessed 10-October-2022].
- S. Ali and M. Patterson, “Spike2vec: An efficient and scalable embedding approach for covid-19 spike sequences,” in *IEEE International Conference on Big Data (Big Data)*, 2021, pp. 1533–1540.
- R. Singh, A. Sekhon, K. Kowsari, J. Lanchantin, B. Wang, and Y. Qi, “Gakco: a fast gapped k-mer string kernel using counting,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017, pp. 356–373.
- S. Ali, B. Bello, P. Chourasia, R. T. Punathil, Y. Zhou, and M. Patterson, “Pwm2vec: An efficient embedding approach for viral host specification from coronavirus spike sequences,” *MDPI Biology*, 2022.