Supplementary Information for Multiscale Grassmann Manifolds for Single-Cell Data Analysis

Xiang Xiang Wang[1], Sean Cottrell[1,2], and Guo-Wei Wei[1,3,4*]

[1] Department of Mathematics,
Michigan State University, East Lansing, MI 48824, USA.
[2] Department of Computational Mathematics, Science, and Engineering,
Michigan State University, East Lansing, MI 48824, USA.
[3] Department of Biochemistry and Molecular Biology,
Michigan State University, East Lansing, MI 48824, USA.
[4] Department of Electrical and Computer Engineering,
Michigan State University, East Lansing, MI 48824, USA.

November 12, 2025

Contents

*Corresponding author. Email: weig@msu.edu

1 Experimental Setup and Parameter Configurations

For each single-cell dataset, let M denote the number of samples (cells), and let $X \in \mathbb{R}^{M\times N}$ represent the preprocessed gene expression matrix. Each dataset is first reduced to N_{PCA} dimensions by PCA, followed by multi-scale UMAP embeddings with output dimension $N_{\mathrm{UMAP}} = n$. Across multiple neighborhood sizes, the resulting UMAP embeddings form a set of p-dimensional subspaces in $\mathbb{R}^n$, allowing each cell to be represented as a point on the Grassmann manifold $\mathbf{Gr}(n, p)$.

The neighborhood scales are generated using a power-based sampling rule that controls the distribution of sampled scales between two bounds $s_{\min} = a$ and $s_{\max} = b$. Given a total of n_{scales} candidate scales and an exponent $p_s > 0$, the continuous scale sequence is computed as

$$\tilde{s}_i = a + (b-a)\left(\frac{i}{n_{\mathrm{scales}}-1}\right)^{p_s}, \qquad i = 0, \ldots, n_{\mathrm{scales}} - 1, \tag{1}$$

which is then rounded to integers and deduplicated to obtain the final ordered set $S = \{s_1 < \cdots < s_p\}$ with $p \le n_{\mathrm{scales}}$. In this work, Setup I adopts $p_s = 2.0$ (`power2`) and Setup II adopts $p_s = 1.6$ (`power1.6`).

1.1 Setup I (power2 sampling)

Table 1 summarizes the configuration of Setup I, where neighborhood sizes are generated using a power-law sampling rule with exponent $p_s = 2.0$. Each dataset produces Grassmann manifold representations $\mathbf{Gr}(n, p)$ under this configuration, where n is the UMAP embedding dimension.

Table 1: Parameter configuration for Setup I (`power2`). Each dataset forms Grassmann manifold representations $\mathbf{Gr}(n, p)$.

Dataset	N	N_{PCA}	n	n_{scales}	p	Scales $(s_1, \ldots, s_p)$
GSE75748time	758	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE94820	1140	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE67835	420	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE75748cell	1018	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE109979	328	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE84133Human1	1895	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE84133Human2	1702	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE84133Human4	1275	200	100	25	23	5,6,8,9,11,13,16,18,21,25,29,33,37,42,47,53,58,65,71,78,85,92,100
GSE57249	49	40	30	13	12	5,6,7,8,9,11,14,16,19,22,26,30

Note. In this setup, the sampling function $s_i = a + (b-a)(i/(n_{\mathrm{scales}}-1))^{2.0}$ is used to generate n_{scales} candidate neighborhood sizes, where $a = s_{\min} = 5$ and $b = s_{\max} = n$. After rounding and deduplication, the number of unique scales $p \le n_{\mathrm{scales}}$ defines the subspace dimension in $\mathbf{Gr}(n, p)$.

1.2 Setup II (power1.6 sampling)

Setup II adopts a milder exponent ($p_s = 1.6$) in the power-based sampling function, producing denser coverage of intermediate neighborhood ranges.

Each dataset generates subspaces on $\mathbf{Gr}(n, p)$, where n is the UMAP embedding dimension. The detailed parameter configurations for this setup are summarized in Table 2.

Note. The sampling rule $s_i = a + (b-a)(i/(n_{\text{scales}}-1))^{1.6}$ is applied with $a = s_{\min} = 5$ and $b = s_{\max} = n$. After rounding and removing duplicates, the resulting number of distinct scales $p \leq n_{\text{scales}}$ defines the subspace dimension in $\mathbf{Gr}(n, p)$. Across all M datasets, these configurations ensure consistent representation dimensions and balanced multi-scale coverage.

Table 2: Parameter configuration for Setup II (`power1.6`). Each dataset forms Grassmann manifold representations $\mathbf{Gr}(n, p)$.

Dataset	N	N_{PCA}	n	n_{scales}	p	Scales $(s_1, \ldots, s_p)$
GSE75748time	758	50	20	11	10	5,6,7,8,10,12,13,15,18,20
GSE94820	1140	50	20	11	10	5,6,7,8,10,12,13,15,18,20
GSE67835	420	50	20	11	10	5,6,7,8,10,12,13,15,18,20
GSE75748cell	1018	50	20	11	10	5,6,7,8,10,12,13,15,18,20
GSE109979	328	50	20	11	10	5,6,7,8,10,12,13,15,18,20
GSE84133Human4	1275	100	50	20	19	5,6,7,9,10,12,14,16,19,21,24,27,30,33,36,39,43,46,50
GSE57249	49	20	15	10	9	5,6,7,8,9,10,12,13,15
GSE84133Human1	1895	100	50	20	19	5,6,7,9,10,12,14,16,19,21,24,27,30,33,36,39,43,46,50
GSE84133Human2	1702	100	50	20	19	5,6,7,9,10,12,14,16,19,21,24,27,30,33,36,39,43,46,50

2 Robustness Analysis under Scale Variation

To evaluate the robustness of the proposed MGM framework under different scale conditions, we examined two representative datasets exhibiting distinct scale behaviors. The first case (GSE57249) demonstrates the stability of MGM when single-scale UMAP embeddings already perform consistently well across scales, while the second case (GSE84133Human2) highlights its resilience when multiple scales contain noisy or weak features. Together, these results illustrate that Grassmann manifold aggregation enables both stable and adaptive representations under varying scale quality.

2.1 Case 1: Stability under uniform scales (GSE57249)

In this dataset, individual PCA→UMAP embeddings already achieve high clustering performance across neighborhood sizes. As shown in Table 3, single-scale UMAP results remain stable for scales between 5 and 10, with a gradual decline for larger neighborhoods. MGM maintains a consistently high accuracy comparable to the best-performing single-scale embedding, demonstrating that the Grassmann manifold integration does not dilute discriminative information when all scales are of similar quality.

Table 3: Clustering performance under different neighborhood scales for GSE57249 (Setup II, seed = 1). MGM achieves the same performance as the best single-scale configuration, confirming its stability under uniformly strong scales.

Method	Scale (s)	ACC	ARI	NMI	Purity
PCA→UMAP	5	0.9796	0.9483	0.9293	0.9796
	6	0.9796	0.9483	0.9293	0.9796
	7	0.9796	0.9483	0.9293	0.9796
	8	0.9796	0.9483	0.9293	0.9796
	9	0.9796	0.9483	0.9293	0.9796
	10	0.9796	0.9483	0.9293	0.9796
	12	0.9592	0.8996	0.8609	0.9592
	13	0.9592	0.8996	0.8609	0.9592
	15	0.9592	0.8996	0.8609	0.9592
Average PCA→UMAP	–	0.9728	0.9320	0.9065	0.9728
MGM (multi-scale)	–	**0.9796**	**0.9483**	**0.9293**	**0.9796**

To visualize the uniformity of performance across neighborhood sizes, Figure 1 presents the four evaluation

metrics as continuous trends over the tested scales. This figure highlights the stability of PCA→UMAP results in this dataset and shows that MGM faithfully retains the strongest performance without being affected by redundant or overlapping scales.

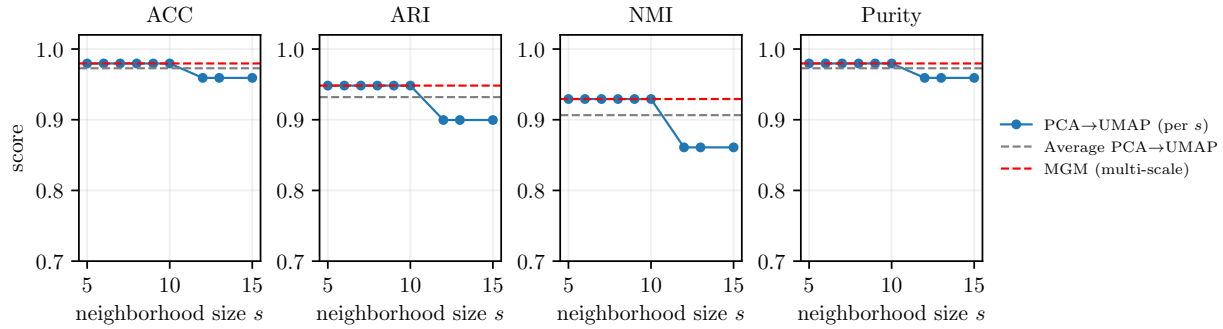

Figure 1: Performance variation across neighborhood scales for GSE57249. Each panel shows one clustering metric (ACC, ARI, NMI, and Purity). Blue solid lines indicate PCA→UMAP evaluated at each scale, gray dashed lines show the mean baseline performance, and the red dashed line denotes MGM (multi-scale). MGM achieves comparable or better results than any single-scale embedding, illustrating its stability under uniformly strong scales.

Observation. When all neighborhood scales are comparably effective, MGM preserves the optimal structure while maintaining numerical stability. This confirms that the multi-scale Grassmann manifold representation does not introduce unnecessary distortion under favorable conditions.

2.2 Case 2: Resilience under noisy scales (GSE84133Human2)

In the GSE84133Human2 dataset (Setup I, seed = 1), the single-scale PCA→UMAP results vary substantially across neighborhood sizes, with several scales exhibiting degraded clustering performance.

Table 4 summarizes the detailed results. Despite strong scale interference, MGM achieves higher accuracy than both the average and the best single-scale UMAP configuration, illustrating its ability to extract and amplify useful geometric features from a mixture of strong and weak subspaces.

To further visualize the variation and robustness across scales, Figure 2 plots all four evaluation metrics as a function of neighborhood size. This comparison highlights how MGM maintains consistently high scores even when single-scale UMAP embeddings deteriorate.

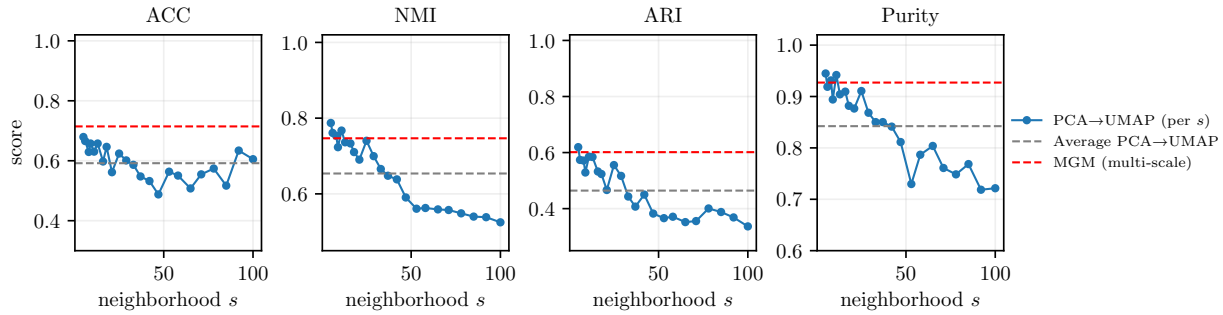

Figure 2: Performance variation across neighborhood scales for GSE84133Human2. Each panel corresponds to one metric (ACC, ARI, NMI, Purity). The blue solid line shows PCA→UMAP performance at each scale, while gray and red dashed lines denote the average PCA→UMAP and multi-scale MGM results, respectively. Even when several scales yield degraded single-scale embeddings, MGM maintains a high, stable performance by leveraging information from multiple subspaces.

Observation. In the presence of scale interference, where several UMAP neighborhoods yield poor embeddings, MGM successfully integrates multi-scale subspaces on the Grassmann manifold to enhance global structure. This demonstrates the ability of Grassmann manifold learning to improve robustness and stability in real-world single-cell datasets.

Table 4: Clustering performance under different neighborhood scales for GSE84133Human2 (Setup I, seed = 1). Although single-scale embeddings fluctuate widely, MGM aggregates them on the Grassmann manifold to restore a stable and discriminative representation.

Method	Scale (s)	ACC	ARI	NMI	Purity
PCA→UMAP	5	0.6792	**0.6194**	**0.7873**	**0.9448**
	6	0.6657	0.5738	0.7610	0.9189
	8	0.6293	0.5714	0.7534	0.9313
	9	0.6575	0.5289	0.7233	0.8942
	11	0.6304	0.5853	0.7672	0.9418
	13	0.6575	0.5842	0.7362	0.9042
	16	0.5975	0.5325	0.7332	0.9095
	18	0.6463	0.5235	0.7104	0.8819
	21	0.5617	0.4665	0.6904	0.8766
	25	0.6240	0.5554	0.7402	0.9107
	29	0.6011	0.5168	0.6996	0.8684
	33	0.5864	0.4432	0.6657	0.8502
	37	0.5476	0.4072	0.6479	0.8502
	42	0.5323	0.4501	0.6383	0.8414
	47	0.4877	0.3824	0.5905	0.8114
	53	0.5635	0.3660	0.5608	0.7297
	58	0.5505	0.3711	0.5627	0.7867
	65	0.5076	0.3519	0.5590	0.8038
	71	0.5546	0.3556	0.5574	0.7609
	78	0.5740	0.4006	0.5486	0.7485
	85	0.5170	0.3879	0.5400	0.7685
	92	0.6340	0.3686	0.5384	0.7186
	100	0.6052	0.3366	0.5249	0.7215
Average PCA→UMAP	–	0.5918	0.4643	0.6538	0.8423
MGM (multi-scale)	–	**0.7145**	0.6015	0.7468	0.9271

2.3 Overall remark

The two case studies together illustrate the adaptability of MGM across different scale conditions. Under uniformly high-quality scales (GSE57249), MGM performs comparably to the best single-scale embedding, maintaining stability without degradation. However, in noisy environments (GSE84133Human2), where several scales yield poor embeddings, MGM exhibits a substantially larger improvement relative to the average single-scale UMAP baseline.

Table 5: Relative improvement (%) of MGM over the average PCA→UMAP baseline in uniform and noisy scale conditions.

Dataset	Condition	ACC↑	NMI↑	ARI↑	Purity↑
GSE57249	Uniform scales	+0.70%	+2.51%	+1.75%	+0.70%
GSE84133Human2	Noisy scales	**+20.7%**	**+14.2%**	**+29.6%**	**+10.1%**

These results indicate that MGM not only preserves performance in favorable conditions but also provides significant robustness benefits when scale quality is heterogeneous. The relative gains are especially pronounced for NMI and ARI under noisy conditions, confirming that Grassmann manifold integration effectively aggregates complementary information from weak subspaces and mitigates scale-specific degradation.

It should be noted that these findings are based on two representative cases, yet they already highlight the broader potential of Grassmann manifold learning. While not every dataset exhibits such large gains, the consistent robustness across varying conditions suggests that the framework can be further optimized, partic-

ularly regarding adaptive selection and weighting of neighborhood scales. This motivates future exploration into data-driven scale modeling to enhance the efficiency and generalization of Grassmann manifold-based representations in large-scale single-cell analysis.

3 Uniform Manifold Approximation and Projection (UMAP)

In this work, the Uniform Manifold Approximation and Projection (UMAP) algorithm [1] is used as the dimension reduction method for generating multiscale embeddings. The resulting Grassmann manifold framework based on these embeddings is referred to as MGM.

Given a dataset $X = \{x_i\}_{i=1}^{M} \subset \mathbb{R}^d$, UMAP constructs a weighted k-nearest neighbor graph that models the local manifold structure of the data. Each edge weight is computed as

$$w_{ij} = \exp\left(-\frac{\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i}\right),$$

where ρ_i denotes the distance from x_i to its nearest neighbor and σ_i is a local normalization parameter. The key hyperparameter `n_neighbors` determines the number of neighbors in the graph and thus controls the scale of locality considered in the embedding.

To obtain multiscale representations, UMAP was applied repeatedly under different neighborhood sizes $s_1, s_2, \ldots, s_p$, producing a sequence of embeddings

$$E_i = \mathrm{MDR}(X; s_i) \in \mathbb{R}^{M \times n}, \quad i = 1, 2, \ldots, p,$$

where each row of E_i represents the n-dimensional embedding of a single cell under scale s_i. Smaller values of s_i emphasize fine-grained local transitions, while larger values capture global structure and lineage organization.

Each multiscale embedding E_i is treated as a distinct feature view of the data. The collection $\{E_1, E_2, \ldots, E_p\}$ is then geometrically integrated into a unified representation on the Grassmann manifold $\mathbf{Gr}(n, p)$. This construction allows MGM to encode multiscale manifold information while maintaining geometric consistency across scales.

4 Clustering Settings

For downstream clustering, two standard algorithms were used depending on the experimental setup. Spectral clustering [5] was applied under the noisy condition (Setup I), and k-means clustering [4] was used under the refined condition (Setup II). Both methods were implemented in `Python` (version 3.11) using the `scikit-learn` package (version 1.3.2). The implementations were adapted for use with precomputed Grassmann manifold distance matrices, and all unspecified parameters followed the library defaults.

4.1 Spectral Clustering (Setup I)

Spectral clustering was performed on a precomputed Grassmann manifold distance matrix D. To convert the distance matrix into an affinity matrix, we used a Gaussian (RBF) kernel of the form

$$A_{ij} = \exp\left(-\frac{D_{ij}^2}{\gamma\, \sigma^2}\right),$$

where γ is a scaling parameter (default = 1.0) and σ^2 was set to the median of all positive squared distances. This produces a symmetric positive similarity matrix that serves as the input to the `SpectralClustering` module in `scikit-learn`, using the `affinity='precomputed'` option and `assign_labels='kmeans'`. The core implementation is summarized below:

```
def spectral_cluster_from_dist(D, n_clusters, gamma=1.0, random_state=1):
    """Spectral clustering on a precomputed distance matrix via RBF affinity."""
    D2 = (D**2).astype(np.float64, copy=False)
    pos = D2[D2 > 0]
    sigma2 = float(np.median(pos)) if pos.size > 0 else 1.0
    if not np.isfinite(sigma2) or sigma2 <= 0:
        sigma2 = 1.0
    A = np.exp(-D2 / (gamma * sigma2))
    sp = SpectralClustering(
        n_clusters=n_clusters,
        affinity='precomputed',
        assign_labels='kmeans',
        random_state=random_state,
        n_init=10
    )
    labels = sp.fit_predict(A)
    return labels, sigma2
```

This implementation follows the general principle of spectral clustering but is not intended as an exact reproduction of any specific algorithmic formulation in the cited literature. It was selected for its simplicity, stability, and compatibility with precomputed distances.

4.2 k-Means Clustering (Setup II)

Under the refined condition, k-means clustering was applied to a classical multidimensional scaling (MDS) embedding derived from the Grassmann manifold distance matrix. The embedding maps the distances into a low-dimensional Euclidean space (up to 20 dimensions), after which standard k-means clustering is performed. The implementation is shown below:

```
def kmeans_from_dist(D, n_clusters, random_state=1):
    """K-Means on classical MDS embedding of D."""
    Z = classical_mds_from_dist(D, target_dim=min(20, D.shape[0]-1))
    km = KMeans(n_clusters=n_clusters, n_init=20, random_state=random_state)
    labels = km.fit_predict(Z)
    return labels
```

Both clustering algorithms used the same random seed to ensure reproducibility. The number of clusters k was determined according to the ground truth labels of each dataset. Unless otherwise noted, the Grassmann manifold chordal distance was used as the default metric.

5 Evaluation Metrics of Clustering

This section defines the four evaluation metrics used in our experiments: clustering accuracy (ACC) [2], normalized mutual information (NMI) [8], adjusted Rand index (ARI) [3], purity [6] and average-purity (Avg-Purity) [7]. Let $\{y_i\}_{i=1}^N$ and $\{c_i\}_{i=1}^N$ denote the reference and predicted labels of the N samples, respectively. Each metric evaluates clustering performance from a different perspective, providing complementary measures of accuracy and consistency.

5.1 Clustering Accuracy (ACC)

The clustering accuracy (ACC) measures the proportion of correctly classified samples after aligning predicted cluster labels with the reference labels. Formally,

$$\mathrm{ACC} = \frac{1}{N}\sum_{i=1}^{N}\delta(y_i, f(c_i)),$$

where $\delta(a,b) = 1$ if $a = b$ and 0 otherwise, and $f(\cdot)$ represents the optimal one-to-one mapping between cluster labels and reference labels, typically determined using the Hungarian algorithm [2] . ACC values range from 0 to 1, where higher values indicate stronger agreement between the clustering results and the reference labels.

5.2 Normalized Mutual Information (NMI)

The normalized mutual information (NMI) [8] quantifies the shared information between two label assignments, normalized by their individual entropies:

$$\mathrm{NMI} = \frac{2\,I(Y;C)}{H(Y) + H(C)},$$

where $I(Y;C)$ denotes the mutual information between the reference partition Y and the predicted partition C, and $H(\cdot)$ represents the entropy. NMI ranges from 0 to 1, where 1 corresponds to perfect correlation between the two clusterings, and 0 indicates complete independence.

5.3 Adjusted Rand Index (ARI)

The adjusted Rand index (ARI) [3] evaluates the similarity between two partitions by examining all pairs of samples and determining whether they are assigned to the same or different clusters in both. Let $Y = \{Y_1, \ldots, Y_L\}$ and $C = \{C_1, \ldots, C_L\}$ denote the sets of reference and predicted clusters, respectively. Define $n_{ij} = |Y_i \cap C_j|$ as the number of samples belonging to both class i and cluster j, $a_i = \sum_j n_{ij}$, and $b_j = \sum_i n_{ij}$. The ARI is computed as:

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\Big/\binom{N}{2}\right]}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\Big/\binom{N}{2}\right]}.$$

ARI ranges from -1 to 1, where 1 indicates perfect agreement, 0 corresponds to random labeling, and negative values indicate less agreement than expected by chance.

5.4 Purity

Purity [6] measures the homogeneity of each predicted cluster in relation to the reference labels. For each predicted cluster C_k, the class that contributes the largest number of samples is identified. Summing over all clusters gives:

$$\mathrm{Purity} = \frac{1}{N}\sum_k \max_j |C_k \cap Y_j|.$$

Purity values range from 0 to 1, where 1 indicates that each predicted cluster contains samples from only one reference class. Unlike ACC, purity does not rely on label matching but instead evaluates cluster composition directly.

5.5 Average Purity

The average-purity (Avg-Purity) metric [7] is used to assess the consistency of clustering assignments by quantifying how concentrated the samples are within each predicted cluster. Let S be a finite set of $n = |S|$ samples, and let $\mathcal{P} = \{S_1, S_2, \ldots, S_k\}$ be a partition of S into k disjoint subsets such that $\bigcup_{i=1}^{k} S_i = S$. The purity of the partition $\mathcal{P}$ is defined as

$$\text{purity}(\mathcal{P}) = \sum_{i=1}^{k} \left(\frac{|S_i|}{n} \right)^2. \tag{2}$$

A higher purity value indicates that the partition is dominated by a few large homogeneous subsets, whereas a lower value suggests a more even or fragmented distribution. To assess the overall taxonomic or class consistency, the average-purity of all labels $\mathcal{L}$ in the dataset is computed as

$$\text{Avg-Purity} = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \text{purity}(\mathcal{P}_l), \tag{3}$$

where $\mathcal{P}_l$ denotes the partition of samples corresponding to label l. A value of Avg-Purity close to 1 indicates strong intra-class coherence across all clusters.

References

[1] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019.

[2] David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

[3] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[4] James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, pages 281–297, 1967.

[5] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[6] KVSN Rama Rao and B Manjula Josephine. Exploring the impact of optimal clusters on cluster purity. In *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, pages 754–757. IEEE, 2018.

[7] Faisal Suwayyid, Yuta Hozumi, Hongsong Feng, Mushal Zia, JunJie Wee, and Guo-Wei Wei. Cakl: Commutative algebra k-mer learning of genomics. *arXiv preprint arXiv:2508.09406*, 2025.

[8] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080, 2009.