p-ClustVal: A Novel p-adic Approach for Enhanced Clustering of High-Dimensional scRNASeq Data

Parichit Sharma[1,2*], Sarthak Mishra[2], Hasan Kurban[2,3,4,5], Mehmet Dalkilic[2,3]

[1]Luddy Center for Artificial Intelligence, Bloomington, 47408, Indiana, USA.
[2]Department of Computer Science, Luddy School of Informatics, Computing & Engineering, Bloomington, 47408, Indiana, USA.
[3]Department of Data Science, Luddy School of Informatics, Computing & Engineering, Bloomington, 47408, Indiana, USA.
[4]Electrical and Computer Engineering Department, Texas A&M University at Qatar, Doha, 23874, Doha, Qatar.
[4]College of Science and Engineering, Hamad Bin Khalifa University, Doha, 23874, Doha, Qatar.

*Corresponding author(s). E-mail(s): parishar@iu.edu;
Contributing authors: samishr@iu.edu; hkurban@hbku.edu.qa; dalkilic@iu.edu;

Abstract

This paper introduces ***p***-ClustVal, a novel data transformation technique inspired by ***p***-adic number theory that significantly enhances cluster discernibility in genomics data, specifically Single Cell RNA Sequencing (scRNASeq). By leveraging ***p***-adic-valuation, ***p***-ClustVal integrates with and augments widely used clustering algorithms and dimension reduction techniques, amplifying their effectiveness in discovering meaningful structure from data. The transformation uses a data-centric heuristic to determine optimal parameters, without relying on ground truth labels, making it more user-friendly. ***p***-ClustVal reduces overlap between clusters by employing alternate metric spaces inspired by ***p***-adic-valuation, a significant shift from conventional methods. Our comprehensive evaluation spanning 30 experiments and over 1200 observations, shows that ***p***-ClustVal improves performance in 91% of cases, and boosts the performance of classical and state of the art (SOTA) methods. This work contributes to data analytics and genomics by introducing a unique data transformation approach, enhancing downstream clustering algorithms, and providing empirical evidence of ***p***-ClustVal's efficacy. The study concludes with insights into the limitations of ***p***-ClustVal and future research directions.

Keywords: Data-Centric AI, Unsupervised Learning, ***p***-Adic Numbers, Clustering High Dimensional Data, Single Cell RNA Sequencing

1 Introduction

Clustering is a pivotal technique that organizes data into meaningful groups, unveiling inherent structures without prior labeling. It is especially critical in real-world applications, like biological data interpretation, where it helps in identifying distinct patterns and groups based on similarities among data points. However, the increasing dimensionality and sparsity of modern datasets

pose significant challenges to traditional clustering methods. High-dimensional spaces often dilute the effectiveness of conventional distance metrics and exacerbate the curse of dimensionality, leading to suboptimal clustering performance [1].

In recent times, scRNASeq has marked a paradigm shift, enabling an in-depth exploration of data at an unprecedented single-cell resolution [2]. This technological leap has been instrumental in elucidating complex biological processes, providing vital insights across diverse areas in biology [3, 4, 5, 6]. A primary challenge in scRNASeq data analysis is identifying distinct cell groups, akin to unsupervised clustering in machine learning [7]. The field has seen diverse clustering methodologies evolve, each with its own unique approach and data processing strategy [8, 9].

Despite its potential, scRNASeq data processing is beset with challenges, primarily due to the noisy, low-resolution nature of the raw data, typically represented as an $m \times d$ matrix of cells (samples) and gene counts (features). For instance, gene expression is often captured across thousands of genes (dimensions), clustering such high dimensional space is difficult due to distance measures becoming less meaningful in distinguishing data points [1]. scRANSeq data is characteristically sparse i.e. several genes have zero/near-zero counts across the cells, thus making it difficult to find meaningful structures, or establish similarity between data points [3]. Moreover, noise is introduced in the form of dropout events (when a gene expression is not captured even when it's present), due to technical limitation and biological variation [8]. This makes it difficult for clustering algorithms to separate true gene expression from nuisance variables. Due to biological variability (cell development stages, cell cycle variability etc.), there is sufficient variation within cells that can blur boundaries between cell types and states, making clustering more challenging [2]. Consequently, existing clustering algorithms often struggle to effectively cluster such complex data spaces. For completeness, we share a general overview of the standard data acquisition and processing pipeline in Figure 4.

In this work, we introduce p-ClustVal, a novel data transformation technique inspired by the p-adic number theory [10]. By leveraging alternate metric spaces based on p-adic-valuation, p-ClustVal enhances cluster discernibility and separation, setting it apart from conventional methods. By operating in the transformed p-adic space, p-ClustVal mitigates cluster overlap, thereby enhancing cluster separation. This transformation consequently augments the efficacy of downstream clustering algorithms, facilitating the elucidation of previously obscured patterns and structures within the data. One of the key strengths of p-ClustVal lies in its ability to seamlessly integrate with and enhance popular clustering algorithms and dimension reduction techniques. Moreover, p-ClustVal is designed to be user-friendly, requiring minimal parameter tuning. It employs a unsupervised heuristic for dynamically selecting optimal parameters without requiring ground truth labels, making it more accessible. To validate the effectiveness of p-ClustVal in real-world applications, we apply it on several high-dimensional, and sparse datasets in scRNASeq datasets.

Extensive evaluation across 30 experiments comprising 1250 observations reveal that p-ClustVal significantly improves the performance in 91% of cases, with the clustering labels showing a much higher level of agreement with the ground truth. The primary contributions of this work include:

- A novel transformation (p-ClustVal) designed to improve cluster identification in high dimensional single cell data.
- Introducing a unsupervised heuristic to find optimal transformation parameters in a data-driven manner, significantly improving adaptability across datasets and reducing manual configuration.
- An intuitive demonstration of p-ClustVal on synthetic and real world data, and discussion of essential metric properties.
- Comprehensive empirical validation of p-ClustVal's efficacy in enhancing classical and state of the art clustering tools.

The remainder of this paper is organized as follows: Section 2 outlines the background and related work. Section 3 elaborates on our proposed method. Section 4 details the experimental setup, and Section 5 presents our findings. Section 6 addresses the current limitations and explores potential solutions. Finally, Section 7 concludes the paper, offering final thoughts and directions for future research.

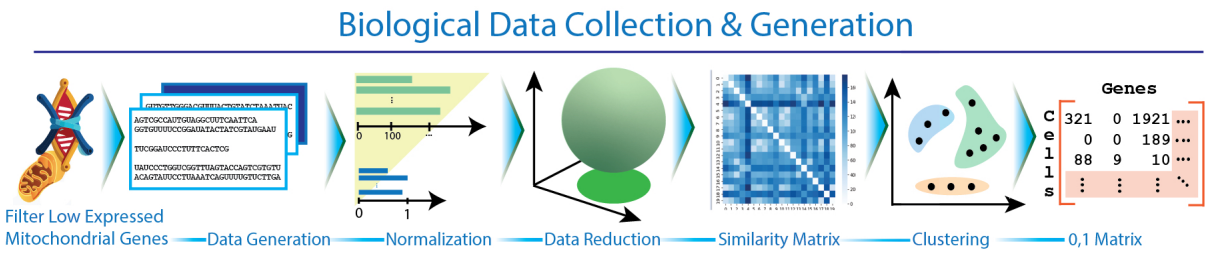

Fig. 1: **Data generation phase**: The samples are extracted from the tissue of interest, followed by cell library preparation and high throughput sequencing. The obtained sequences are aligned with a reference genome. The result is a matrix that stores gene transcripts counts per cell. **Data analysis phase**: The data goes through initial quality control to remove low quality cells and genes, normalization, feature subset selection or dimensionality reduction, building cell similarity matrix either in the lower dimensional or original feature space, and clustering.

2 Related Work

Researchers have employed various techniques to cluster scRNASeq data. Broadly, these can be categorized into 1) partition or hard clustering, 2) hierarchical clustering, 3) graph based and community detection methods, and 4) deep learning based techniques. Most single cell datasets contain inherent noise due to technical factors (variation in sequencing bias, library preparation) and biological variability [11]. Consequently, dimensionality reduction is done to help separate the signal from noise, and mitigate the affect of large number of dimensions. Moreover, clustering is often applied after dimensionality reduction, to reduce the computational cost and avert curse of dimensionality. Here, we summarize the widely used dimensionality reduction techniques, and representative clustering methods, along with their benefits and drawbacks.

Dimensionality reduction techniques

Some of the commonly used methods are: 1) Principal Component Analysis (PCA), 2) T-Distributed Stochastic Neighbor Embedding (t-SNE), and 3) Uniform Manifold Approximation and Projection (UMAP). While being used as standalone tools for reducing dimensions, these methods are often combined with clustering methods to help reduce the impact of noise, improve clustering results and reduce computational cost. Recently, deep neural autoencoder networks have also been used for dimensionality reduction and clustering, these are discussed under clustering methods.

- **PCA** [12] is commonly used to project original data onto a space composed of orthogonal vectors or principal components (PCs). To reduce dimensions, top PCs that capture maximum variance in data are selected for projection. However, PCA assumes linear relationship in the data, and may miss crucial non-linear structure, if present [13].
- **t-SNE** [14] is a non-linear approach for projecting high dimensional data onto a space of low dimensional and non-linear manifolds. It preserves non-linear relationship from high dimensional space by creating a probability distributions over pairs of points in both low and high dimensional space, and minimizing the Kullback-Leibler divergence between the two distributions. it has been successfully applied for dimensionality reduction of single cell datasets [15, 16]. However, t-SNE could be slow on large datasets due to high computational complexity ($O(n^2)$), can yield different results between runs, due to it's stochastic nature, and may not preserve the global structure well.
- **UMAP** [17] is a non-linear technique that aims to preserve both local and global structure, while reducing the dimensionality. It begins by creating a weighted graph of data points in the original space, followed by constructing a fuzzy topological structure representing local and global relationships. It searches for a low dimensional topological embedding having the same structure as that of high dimensional one, and minimize the difference between the two embedding via stochastic gradient descent. It

has been shown to perform better than t-SNE [18]. Though, UMAP preserves local and global structure, and can scale to larger datasets well, the quality of results depend on choice of parameters, and results can also vary between runs due to stochastic nature of the algorithm.

Partition based clustering

In partition based techniques, each data point is exclusively assigned to a single cluster. Classical k-means [19] is an example of partition based clustering. k-means is an iterative algorithm which partitions the data in k clusters, while minimizing the sum of squared errors between data points and centroids. Despite it's popularity and simplicity, k-means has some limitations which affects its application in certain scenarios. For example: sensitivity to initialization, which can sometimes result in sub-optimal solution [20], assumption of spherical clusters which limits it's effectiveness to more complex shapes and non-linear separation [21], requirement to specify the number of clusters, and high computational cost when operating on larger datasets [22]. Consequently, existing tools have integrated k-means with different transformation and distance metrics to enhance it's efficacy. For example, SC3 [23] run multiple trials of k-means on different data transformations (principal components (PCs) and laplacian eigenvectors), to cluster data in the transformed space, and mitigate the impact of sub-optimal initialization.

pcaReduce [24] begins with a large number of clusters and use hierarchical clustering to progressively merge clusters. It apply k-means on PCs rather than raw data, reducing dimensionality and noise. pcaReduce determines the optimal number of clusters by removing one PC at a time, followed by re-clustering of remaining PCs via k-means, and merge most similar clusters, until optimal number of clusters are found. SAIC [25] leverage k-means to identify subset of features, thus aiding in dimensionality reduction and clustering. SAIC also adjusts the number of clusters iteratively, to overcome the k-means's limitation of pre-specifying the number of clusters. SCUBA [26] incorporates temporal aspects to help analyze time series single cell data which traditional k-means can't do. Specifically, it modifies k-means to detect bifurcation events in cell lineages to detect cell development trajectories. Since, euclidean distance become less effective in higher dimensions [1], RaceID3 [27] substitutes k-means with k-medoids [28] clustering which use Pearson correlation metric instead of euclidean distance.

Hierarchical clustering

Hierarchical clustering, which encompasses both agglomerative and divisive methods, organizes data into tree-structured clusters [29]. Agglomerative clustering starts by merging individual data points into progressively larger clusters, culminating in a single cluster at the root of the tree. Conversely, divisive clustering initiates with the entire dataset, systematically dividing it into increasingly finer subgroups. Various computational tools employ these approaches. For example: BackSpin [30] first perform principal component analysis (PCA) to reduce dimensions and mitigate the impact of curse of dimensionality. It starts with all cells as one cluster and iteratively split each cluster into 2 using a modified k-means approach ($k = 2$). CIDR [31] begins with data imputation to address the issue of dropouts (zero values). This helps to overcome k-means' sensitivity to outlier and noise. To address high dimensionality, CIDR perform principal coordinate analysis (PCoA), and clustering is done on top principal coordinates. Notably, SC3 integrate hierarchical clustering with consensus matrices to refine the results obtained from k-means clustering.

Graph & community detection methods

Graph based clustering use graph representation of data, for example by finding nearest neighbors followed by determining closely connected communities (clusters) in the graph. Examples include: Louvian clustering [32] which does not require specifying the number of clusters, a priori, and is well suited to detect communities in large graphs. Seurat [33] is a widely used framework that use Louvian clustering. It starts with reducing the dimensionality of gene expression data, typically via PCA. Thereafter, a K nearest neighbor (KNN) graph is constructed by finding euclidean distance in PCA space, followed by refining the KNN graph into a shared nearest neighbors (SNN) graph (to capture subtle cell-cell relationships). The SNN graph is partitioned into communities or modules via Louvian clustering. Spectral clustering [34]

aims to transform data to a space, where it's easier to discern cluster separation and identify non-spherical clusters. It begins with constructing a similarity graph from data (where, each node represents a data point), graph laplacian is computed from the similarity graph, followed by k-means on top eigenvectors of the graph laplacian. SIMLR [35] use multiple kernels to assess cell-cell similarities via different kernel metrics. This allows learning data specific metric, making it robust to noise and heterogeneity in data, and surpass limitation of using predefined similarity metrics.

Deep learning approaches

Recently, deep learning based approaches have gained significance for clustering single cell datasets. Examples include, scDeepCluster [36] combines a deep autoencoder network with a dedicated clustering layer to simultaneously optimize reconstruction and clustering loss. It employ a self-training mechanism by using clustering labels as pseudo labels to iteratively train the network and refine clustering results. By end-to-end combining the autoencoder, clustering, and imputation layer, scDeepCluster addresses the issues of high dimensionality and dropouts while learning a latent representation fit for clustering. scziDesk [37] which use zero inflated negative binomial (ZINB) distribution to explicitly model the dropout events in the data. It employs a deep autoencoder architecture with modified ZINB loss function to learn low dimensional representation of data. scziDesk introduced a discriminative component in the network loss function to help differentiating between cells. By combining ZINB distribution with the deep autoencoder, and discrimination aware loss function, scziDesk address key challenges such as: dimensionality, sparsity and dropouts. ADClust [38] beings with learning the low dimensional embedding via pre-trained autoencoders, and partition the embedding into micro-clusters. The micro-clusters are compared via statistical tests and similar clusters are merged into larger ones. The advantage of ADClust is automatic determine of optimal number of clusters, and it does not require the number of clusters be pre-specified.

Miscellaneous

Weighted ensemble methods attempt to improve cell heterogeneity analysis by integrating multiple clustering results, often weighted by the reliability of base clustering. SAFE [39] combines four state-of-the-art clustering techniques (SC3, CIDR, Seurat, t-SNE + k-means), and use hyper-graph partitioning to ensemble the results into a consensus matrix. sc-GPE [40] combines graph based clustering techniques to construct the consensus matrix, followed by hierarchical clustering on consensus matrix. scEWE [41] proposed higher-order element-wise weighting strategy, to assign different clustering weights to different cells, leading to a robust consensus matrix. scEWE mitigates the issue of ignoring low-level structure information (cell-cell relationships) in previous ensemble methods by assigning different weights to cells.

jSRC [42] perform joint optimization of dimension reduction and clustering to address scalability and clustering accuracy. [43] use LDA (latent Dirichilet Allocation) to identify genes representative of various clusters, thus enabling data-driven cluster annotation. SCENA [44] select specific genes (feature set), followed by improving local affinity amongst cells and clustering based on a consensus matrix. FEATS [45] focus on selecting top informative features for downstream clustering, cluster number annotation and outlier detection. scHFC [46] combine Fuzzy C-means (FCM) and Gath-Geva (GG) algorithms with simulated annealing and genetic algorithm for clustering the scRNASeq data.

Application of p-adic Theory

The concept of p-adic numbers is fundamental to number theory and has recently been applied in diverse scientific fields, including genetics and computer science. For example, [47] utilized the hierarchical properties of p-adic spaces to develop a framework for interpreting network structures. They employed p-adic numbers in Erdös-Rényi graphs to simulate giant components relevant to fields such as genetics, social networks, and epidemics. [48] explored p-adic numbers in the analysis of the standard genetic code and vertebrate mitochondrial code, using 5-adic and 2-adic distances to assess codon proximity. [49] proposed a connection between p-adic statistical field theory and neural networks, while [50] generalized cellular neural networks using p-adic numbers.

Despite these advances, the integration of p-adic numbers in machine learning is still nascent.

In this study, we draw inspiration from the work of [51] on p-adic vectors and propose an innovative approach to enhance unsupervised clustering. Previous efforts have been made to use the natural hierarchical properties of p-adic numbers for sparse encoding, thus improving clustering [52, 53]. Some authors have also used the p-adic metric instead of Euclidean, Manhattan, Chebyshev, or cosine metrics to improve k-nearest neighbor [54]. Here, we present a different transformation inspired by the p-adic valuation. Our method is able to discretize the space using parameters inferred from the data, thus introducing a novel data representation.

Comparison with existing transformation methods

As discussed earlier, data transformation methods come with their unique strengths and drawbacks. For example, assumption of linearity (PCA) versus non-linearity (t-SNE/UMAP) affects how well a given method can discover complex non-linear relationships [55]. Preservation of global (PCA/UMAP) versus local structures (t-SNE) determines if the results would conserve overall structure or emphasize local fine-grained relationships [18]. Choice of dimensionality reduction method also affect the technical noise and computational cost differently [56]. Since, many clustering tools include dimensionality reduction (PCA, PCoA, deep neural autoencoders etc.), the results are also affected by the choice of algorithm [57]. In contrast, p-ClustVal does not assume linearity/non-linearity, rather than explicitly preserving global vs. local structure, the emphasis is to increase cluster separability in the transformed space. Experiments show that, where performance of dimensionality reduction methods may vary with data, p-ClustVal improves clustering accuracy across data and dimensionality reduction (Sec. 5). This indicates that p-ClustVal is robust to data specific characterstics. Unlike some methods where final results vary due to stochastic nature of the algorithm (t-SNE/UMAP), p-ClustVal follows deterministic process that always yield same results. Moreover, compared to prevalent dimensionality reduction techniques, p-ClustVal is computationally efficient, as it scales well with data size, and relatively cheap to compute. Further details on complexity analysis are given in Sec. 3.7.

3 Methods

We define the notation as follows: let $\mathbf{x} \in \mathbb{R}^d$ represent a d-dimensional vector over the reals. The dataset $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ comprises m such vectors. We denote the ith element of $\mathbf{x}$ by x_i. The distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is given by $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$. The number of centroids is denoted by $k \in \mathbb{N}$, and we also use k to denote the top k neighbors of a data point, with the specific usage clarified by the context.

3.1 p-adic numbers

P-adic numbers were first introduced by Kurt Hensel in 1897. P-adic numbers are an extension of rational numbers obtained by completing the rational numbers with respect to the p-adic metric. They are represented as infinite series $a_0 + a_1 p + a_2 p^2 + \ldots$, where a_i are integers and p is a prime number. Convergence of the series is determined by the size of the powers of p. A p-adic series will converge when the terms become arbitrarily small as the index increases, measured by the p-adic-valuation. p-adic-valuation ($V_p(n)$) [10, 58] is a mathematical function that finds the highest power of a prime number p that divides a given non-zero rational number n. It is defined as follows:

$$V_p(n) = \max\{i \in \mathbb{N} : p^i \mid n\} \tag{1}$$

Given the diverse applications of p-adic numbers in scientific domains, and considering the capability of p-adic-valuation to provide alternative data representations with respect to a prime p, we introduce a parametric space discretization technique inspired by p-adic-valuation. Specifically, our transformation function discretizes data to enhance the separation between different classes while preserving the proximity of data points within the same class.

3.2 p-ClustVal: Addressing the Limitations of p-adic-valuation

In p-adic-valuation, the largest exponent of a prime that divides a number is identified. For instance, the 2-adic valuation of 3 is 0, because

the highest power of 2 that divides 3 is 0 ($2^0 = 1$). Consider a set of data points, $X = \{3, 5, 11, 67, 129, 259\}$. Calculating the 2-adic valuation results in $V_2(X) = \{0, 0, 0, 0, 0, 0\}$, *i.e.,* all data points are mapped to 0, losing the inherent distinctions in the data. Such loss of distinction compromises clustering; for instance, if $k = 2$ and k-means is done on $V_2(X)$, it would erroneously group all points into a single cluster. This is an undesirable outcome, because after the valuation, we would ideally want data points $3, 5, 11$ to be assigned a different cluster than $67, 129, 259$. On the contrary, p-adic-valuation in it's current form would force k-means to assign these points as one cluster. Therefore, to transform a given number x (while preserving relative order of magnitude), we find the highest power of a reference number such that reference raised to that power is smaller than the given number x. We relax the constraint of using prime number, thus allowing the reference to be any real number, and refer to it as the parameter of cohesion (`coh`). We call this new transformation as p-ClustVal. Using this new adaptation and `coh` set to 2, p-ClustVal($X = \{3, 5, 11, 67, 129, 259\}$) $\rightarrow$ $\{1, 2, 3, 6, 7, 8\}$, reflecting that 2^1 is the highest power of 2 which is less than 3, 2^2 is less than 5, 2^3 is less than 11, 2^6 is less than 67, and similarly for other values in X.

With this new valuation, numbers $1, 2, 3$ are closer to each other, similarly $6, 7, 8$ have become closer. If k-means is run on this new valuation, naturally it has higher chances of better clustering because numbers $1, 2, 3$ are further from numbers $6, 7, 8$. Since all the numbers between coh^n to coh^{n+1} will end up at n, our transformation preserves relative monotonicity of datum's magnitude. This enhances the clustering accuracy since the transformed values corresponds better with the magnitude of data.

3.3 Transformation Function

We define a transformation function $\theta : \mathbb{R} \rightarrow Z$, parameterized by separation (`sep`) and cohesion (`coh`), for any scalar $x_i \in \mathbb{R}$. For simplicity, here we denote `coh` by c, and `sep` by s. This function is given by:

$$\theta_{c,s}(x_i) = \lfloor \log_c(x_i^s) \rfloor \tag{2}$$

The transformation is applied element wise to every x_i of vector $\mathbf{x}$, such that s scales each dimension (d) of $\mathbf{x}$, spreading the data points apart, while c maps the scaled data to c^d hypercubes. After the transformation, we further separate the data via uniform expansion ($\theta_{c,s}(\mathbf{x})^2$). Since the transformation is done element wise, the run time complexity is linear in the size of data i.e. $O(md)$.

3.4 Preservation of Distance Metric

Our transformation θ is designed to preserve the properties of the metric space:

Identity of Indiscernibles: For a datum $\mathbf{x}$ in the input space, we have $d(\mathbf{x}, \mathbf{x}) = 0$. This property is preserved post-transformation:

$$d(\theta_{c,s}(\mathbf{x}), \theta_{c,s}(\mathbf{x})) = d\left(\lfloor \log_c(\mathbf{x}^s) \rfloor, \lfloor \log_c(\mathbf{x}^s) \rfloor\right) = 0 \tag{3}$$

Positivity: For datum $\mathbf{x}, \mathbf{y}$, with $d(\mathbf{x}, \mathbf{y}) > 0$, their transformed counterparts also maintain a positive distance:

$$d(\theta_{c,s}(\mathbf{x}), \theta_{c,s}(\mathbf{y})) = d\left(\lfloor \log_c(\mathbf{x}^s) \rfloor, \lfloor \log_c(\mathbf{y}^s) \rfloor\right) > 0 \tag{4}$$

This holds unless $\mathbf{x}$ and $\mathbf{y}$ are mapped to the same point by θ. If $\mathbf{x}$ and $\mathbf{y}$ are mapped to the same point, then distance between them is 0 by Eq. (3).

Symmetry: For datum $\mathbf{x}, \mathbf{y}$ in the input space, satisfying $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$, the transformed space retains this symmetry:

$$\begin{aligned} d(\theta_{c,s}(\mathbf{x}), \theta_{c,s}(\mathbf{y})) &= d(\lfloor \log_c(\mathbf{x}^s) \rfloor, \lfloor \log_c(\mathbf{y}^s) \rfloor) && (5) \\ &= d(\lfloor \log_c(\mathbf{y}^s) \rfloor, \lfloor \log_c(\mathbf{x}^s) \rfloor) && (6) \\ &= d(\theta_{c,s}(\mathbf{y}), \theta_{c,s}(\mathbf{x})) && (7) \end{aligned}$$

The symmetry is maintained as the transformation is consistent for all points.

Triangle Inequality: For $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, if $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$, then:

$$\begin{aligned} & d(\theta_{c,s}(\mathbf{x}), \theta_{c,s}(\mathbf{y})) + d(\theta_{c,s}(\mathbf{y}), \theta_{c,s}(\mathbf{z})) && (8) \\ &= d(\lfloor \log_c(\mathbf{x}^s) \rfloor, \lfloor \log_c(\mathbf{y}^s) \rfloor) + d(\lfloor \log_c(\mathbf{y}^s) \rfloor, \lfloor \log_c(\mathbf{z}^s) \rfloor) && (9) \\ &\geq d(\lfloor \log_c(\mathbf{x}^s) \rfloor, \lfloor \log_c(\mathbf{z}^s) \rfloor) && (10) \end{aligned}$$

This is proven by the monotonic nature of the transformation. Let $\mathbf{n} = \theta_{c,s}(\mathbf{x})$, $\mathbf{m} = \theta_{c,s}(\mathbf{y})$, $\mathbf{z} = \theta_{c,s}(\mathbf{z})$. If $d(\mathbf{0}, \mathbf{y}) > d(\mathbf{0}, \mathbf{x})$, then

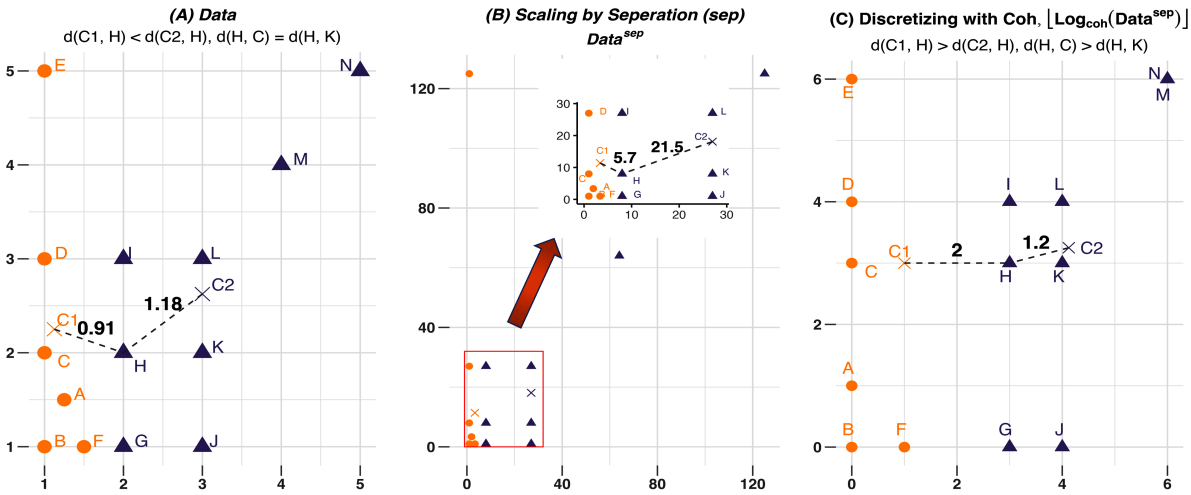

Fig. 2: Visualizing the affect of transformation on a simulated example. Orange and Violet points represent two different clusters. $C1$ and $C2$ are the respective centroids. (A): actual data, (B): after scaling the data with `sep` $= 3$. Figure in the inset highlights the data within the red region. (C): after discretizing with `coh` $= 2$. Dashed lines indicate the distance between cluster centers $C1$, $C2$ and H. Scaling pushes data points apart, but conserves the global spatial structure of data, as seen in plot-(B). Discretizing with `coh` brings data points in the same cluster closer, thus increasing the separation between different clusters.

$d(\mathbf{0}, \lfloor \log_c(\mathbf{y}^s) \rfloor) > d(\mathbf{0}, \lfloor \log_c(\mathbf{x}^s) \rfloor)$. Similarly, for $\mathbf{z}$ and $\mathbf{y}$, and $\mathbf{z}$ and $\mathbf{x}$. Therefore, for transformed points $\mathbf{m}$, $\mathbf{n}$, and $\mathbf{k}$, $d(\mathbf{m}, \mathbf{n}) + d(\mathbf{m}, \mathbf{k}) \geq d(\mathbf{n}, \mathbf{k})$, maintaining the triangle inequality in the transformed space.

3.5 Illustrative examples

Consider the example in Fig. 2-(A), which shows data in two dimensions with points in violet and orange representing two different clusters. Points `G`, `H`, and `I` are equidistant from members of both clusters. Moreover, `H` is closer to `C1` than `C2`, and if the points are assigned based on their proximity to the centers then `H` will be misclassified as part of `C1`. Fig. 2-(B) illustrates the data after scaling by `sep` $= 3$. Though, scaling pushes data points apart, it does not reduce the misclassification risk, because the spatial orientation of data is unaffected. This issue is addressed via discretization with `coh` $= 2$ (Fig. 2-(C)). The transformation relocates boundary points `G`, `H`, and `I` closer to their actual center, significantly enhancing the clustering accuracy. Data points are mapped to the edges of squares of dimension `coh` x `coh`, and some points, for instance: `M, N` are mapped to the same location in the transformed space. This is because for each dimension in the actual data, the transformation process is essentially collapsing all points between coh^n to coh^{n+1} at n. Although, the example is illustrated in two dimensions, same process is extensible to higher dimensions by applying it on each dimension individually. The example intuitively explains the advantage of using p-ClustVal in conjunction with clustering. However, the application is not limited to simulated data, and we show that similar behaviour can be obtained on real world data.

Fig. 3 visualizes the affect of transforming the Muraro dataset (details given in Table 1). The data has 9 classes as shown by the types in the legend. The points are colored by true cluster labels and the circles represent the 85% confidence interval for the data. The first plot (top) shows the actual data, and the second plot (bottom) shows the transformed data on the top two principal components. The overlap between clusters, for instance: `PP cell and A cell`, is noticeably reduced in transformed data. Additionally, for clusters where the spread of the data was high, for example: `Acinar cell, Ductal cell`, the spread is visibly reduced and clusters have become significantly compact after the transformation. Specifically, data points in

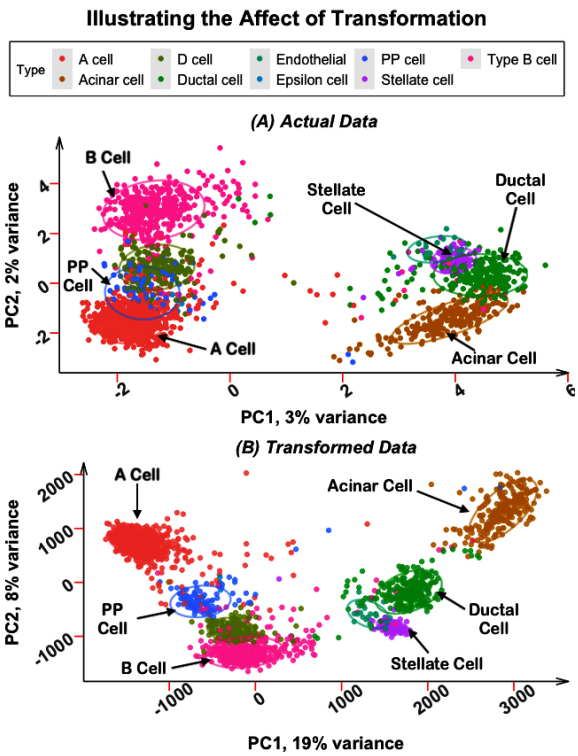

Fig. 3: Visualization of Muraro data on first two principal components. (A): actual data, (B): transformed data. Post transformation, overlap between different clusters has lessened driven by the reduction in spread of data within clusters, and separation between clusters have increased.

cluster `PP cell` have gotten closer while moving away from neighboring clusters. Spread of points in `Stellate cell` has also reduced. Compared to the actual data, the transformation has noticeably improved the chances of clustering `A cell`, `PP cell`, `Stellate cell`, `Acinar cell` and `Ductal cell`. The visual interpretation is also supported by the quantitative assessment on several real world datasets (Sec. 4).

3.6 Data-centric Search for Optimal Parameters

One can understand the effect of p-ClustVal as creating n-dimensional hypercubes of volume $\texttt{coh}^n$ and mapping the data points to the corners of the hypercube. The question is: how to select optimal `coh` without manual intervention, and no access to actual actual class labels? If we make a hypercube too big, we may essentially collapse too many points into the corner of the hypercube, thereby loosing true modality of data. On the contrary, if the hypercube is too small, we may not collapse the closest points together, thereby not compressing excessively spread data and increasing the chances of overlap with neighboring clusters.

If ground truth labels are available, one can find values of `coh` specific to different classes, and set `coh` equal to the maximum intra-class spread. This ensures that all points within a class are mapped to their respective bins, thus significantly improving the chances of grouping all points within the same class as one cluster. However, in most real world data, actual labels are seldom available, and if they are present, then there is no need of clustering. Therefore, to make p-ClustVal work in real world scenarios, we need a heuristic to dynamically find parameters in an unsupervised manner from the data, without access to the ground truth. Our objective is to ensure that p-ClustVal can automatically adapt the parameters search to different datasets, while being user friendly *i.e.,* relaxing the burden of manual parameter tuning. Since `coh` is affected by the overall spread of data, we need to come up with a way to automatically probe the dataset.

A reasonable value of `coh` would be the smallest distance between any two points within the dataset. This is reasonable, as all the closest points must be mapped into the same corner of their respective hypercube. Based on this assertion, we developed a data-centric heuristic for finding the value of `coh` based on the nearest neighbors approach. The closest points are determined by finding the top k nearest neighbors of a given data point. Since, `coh` ensures that closest points are mapped to the same corner, hence, we set `coh` as the average distance of the data point with it's k closest neighbors. This is done for all data points, thus giving a global estimate of `coh` which encapsulates proximity within entire dataset. This is defined in (11).

$$coh = \frac{\sum_{i=1}^{m} \sum_{j=1}^{k} d(x_i, x_j)}{m} \tag{11}$$

k denote the number of neighbors and $x_i \neq x_j$.

Choice of k affects the neighborhood and thereby affect the estimate of `coh`, for example: setting k to a small value would make `coh` smaller,

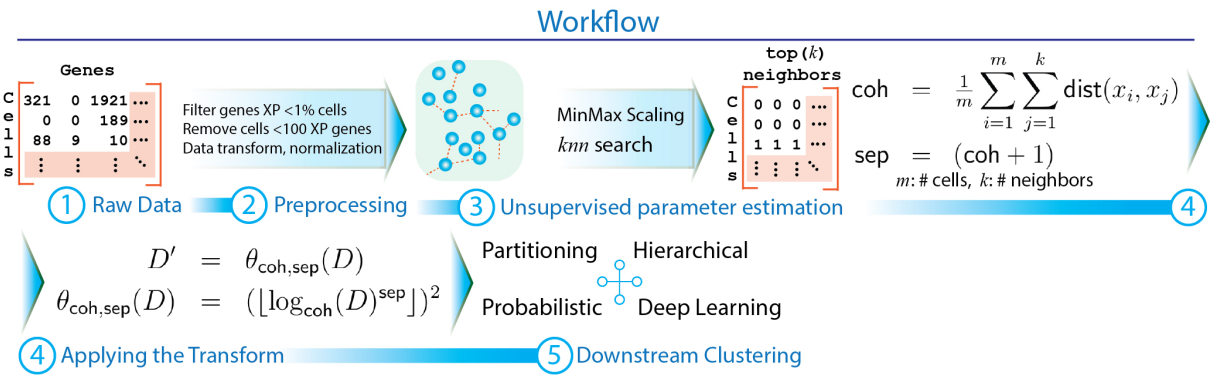

Fig. 4: Algorithmic workflow in p-ClustVal: Data processing, parameter search and transformation.

thus enough points may not get mapped in the same hypercube. On the other hand, if k is too large, *coh* will be inflated, and this will map several points together which might collapse many points from neighboring clusters. We selected the number of neighbors empirically by repeating the clustering on different value of k for 5 randomly selected datasets. Appendix A shows the average accuracy after repeating clustering 20 times on each dataset. The ARI obtained on different k are in similar range, and we arbitrarily choose $k = 5$ for our experiments.

Some features in single cell data can be overly large [59], as a result, distance between closest points can become too large. If *knn* search is done on raw data, it easily inflate the estimate of `coh` and lead to non-numeric results during discretization step (due to using very large value of `coh` as the base of *log*). Therefore, we perform *knn* search on the standardized data, which preserves the original neighborhood [60] and allows to avert the issue of excessively large `coh`. Now, we will discuss about our next parameter *i.e., separation* (`sep`). The goal of separation is to push the points that do not belong in the same cluster further apart. Since, we are using `coh` to estimate proximity in the data, so ideally we want to push the points by at least `coh`+1. By raising the data with `sep` = `coh`+1, we ensure that the space is expanded by at least `coh`. This helps in separating the points that do not belong to the same cluster. Since, `coh` > 0, and `sep` $>$ `coh`, so we are raising data to a positive exponent higher than 1, which will always expand the data space. An illustration of the steps involved in the algorithmic workflow is shown in Figure 4.

3.7 Complexity Analysis

p-ClustVal is an element wise transformation, meaning that it's time complexity is linear in the size of data ($O(md)$ where, m : number of samples and d : number of dimensions). However, the total time is also affected by the parameter search process. The time complexity of parameter search is dominated by the neighbor search ($O(m^2d)$). We obtain the time complexity of our approach by adding the component parts: $O(md) + O(m^2d) \approx O(m^2d)$. Though, total complexity is quadratic, yet p-ClustVal is practically efficient because parameter search is one time process. We benchmarked the time usage of p-ClustVal on large synthetic data [61], and results are shown in Fig. 5. Red line indicate the average time (average of 5 trials) required for p-ClustVal. Across datasets, runtime of p-ClustVal is well below 1 minute. In-fact, on the largest dataset with $100,000$ points, p-ClustVal only takes 57 seconds. The cyan line show the average total time required for executing kNN neighbor search, estimating optimal parameter values and performing the p-ClustVal transformation. On the largest dataset (with 100K points), the total time is 12 minutes. Although, neighbor searches can be executed more efficiently [62, 63, 64], in it's current form, p-ClustVal only takes few minutes to complete the process. We defer the exploration of faster neighbor search to future work.

4 Experiments

Experiments are conducted on 10 high dimensional and sparse scRNASeq datasets, varying in size, dimensions, cluster number, and amount of

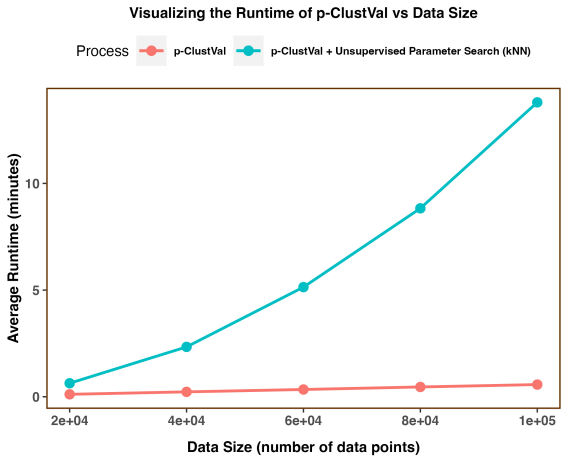

Fig. 5: Scalability analysis on larger datasets. p-ClustVal transform finishes within a minute on all datasets. Although, parameter search is relatively time consuming, but it's a one time process. Total time for neighbor search, parameter estimation and p-ClustVal is few minutes across datasets.

sparsity. Table 1 provide further details. Experiments are designed to underscore the improvement in clustering accuracy by utilizing the transformed versus raw data. To this end, we show that applying the transformation, in addition to improving the accuracy of contemporary clustering methods, it enhances the efficacy of widely used dimension reduction techniques, while also boosting the quality of results in SOTA scRNASeq clustering. As described in Sec. 2, several tools are developed by using the classical clustering algorithms, but these differ greatly in their data processing pipeline, which can also affect the final clustering result, for example, filtering cells or genes below a certain threshold, performing dimension reduction prior to clustering or selecting subset of features for clustering etc. To minimize the impact of processing data differently, first two experiments use same data processing and focus on direct comparison of clustering algorithm that forms the backbone of existing scRNASeq methods. The third experiment focus on comparing domain specific tools, while using their default data processing. Further details on data processing and computing platform are given in appendix C and D.

Table 1: Details of scRNASeq datasets (post processing) including size, features, and cluster count.

Data	m	d	k	Sparsity	Source
Pollen	301	19952	11	72.72	[65]
Darmanis	420	19422	8	78.89	[66]
Usoskin	622	18023	4	80.05	[67]
Mouse Pancreatic	1884	11900	13	86.31	[68]
Muraro	2122	15323	9	66.52	[69]
Limb	1090	16624	6	85.22	[70]
Trachea	1350	18347	4	81.54	[70]
Lung	1676	17243	11	85.23	[70]
Diaphragm	870	15813	5	87.26	[70]
Spleen	9552	9705	5	86.50	[70]

'm': number of data points (or cells), 'd': number of dimensions (or genes), k: true cluster count. Sparsity indicate percentage of zero values in data.

4.1 Benchmark 1: Enhancing Classical Clustering

In these experiments, our objective is to determine whether p-ClustVal can improve the accuracy of classical clustering methods. We specifically demonstrate that data transformed through p-ClustVal and then clustered using a simple algorithm like k-means can outperform more sophisticated algorithms applied to raw data. In the second part of this experiment, we show that p-ClustVal enhances the performance of most algorithms, confirming its compatibility and complementarity with widely used clustering techniques.

We select representative algorithms from the majority of existing techniques, namely k-means, Fuzzy k-means (FZKM) from partition based clustering, expectation-maximization algorithm (EM) for mixture model based clustering and ward agglomerative clustering (WAC) for hierarchical clustering. Here, we limit the scope to classical techniques and SOTA implementations of other techniques viz. graph based clustering, k-medoids clustering, network based community detection and deep learning based clustering are compared in Sec. 5. Results are evaluated over 20 trials and true cluster count for each dataset. For fairness, all algorithms are initialized with same centroids (k++ initialization) in a given trial, and boxplots in Fig. 6 show summary statistics. Average statistics are reported in Table 2.

4.2 Benchmark 2: Clustering with Dimension Reduction

Due to high dimensionality of data, dimension reduction is commonly used either as a preprocessing step to filter noise or to obtain a lower dimensional representation of data before clustering [4, 31, 71]. Here, we want to check if performance can be improved by applying dimension reduction on transformed data. Three frequently used algorithms are compared namely, Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) [14], and Uniform Manifold Approximation and Projection (UMAP) [17]. Results demonstrate that on majority of datasets, applying dimension reduction on transformed data greatly improves the clustering accuracy as reported in Sec.5. For PCA, number of components are selected to capture 95% variance in data, for t-SNE and UMAP-the components are fixed at 2. After applying dimension reduction to raw and transformed data, 20 trials of k-means clustering are done, keeping centroids (k++ initialization) same in a given trial, and average statistics are reported.

4.3 Benchmark 3: Enhancing the state-of-the-art in Domain Specific Clustering

Broadly speaking, available tools for clustering single cell data include packages based on classical clustering or it's variant, packages that perform clustering after projecting data to lower dimensional space, and approaches based on deep learning models to extract latent embedding for clustering. Sheer number of tools make it difficult to do exhaustive comparison, therefore evaluation is done via representative approaches in each category. We compare popular packages namely, Seurat [33] uses network based community detection, RaceID (k-Medoids clustering) [27], SIMLR [72] uses spectral clustering, SC3 [23] combines dimension reduction with multiple matrices (euclidean, pearson coefficient etc.), ADClust [38] that perform clustering on low dimensional representation obtained from auto encoding deep neural networks.

4.4 Evaluation criteria

Quality of clustering assignments is evaluated via Adjusted Rand Index (ARI). Specifically, clustering assignments are compared with ground truth, and ARI is reported. ARI measure the extant of similarity between different clustering assignments and it's value range from -0.5 to 1, where 1 indicates identical clustering, 0 represents random clustering, and negative values indicate less agreement than expected by chance.

5 Results & Discussion

p-ClustVal improves k-means

Results (Fig. 6) show that k-means on transformed data (T-k-means) performs significantly better than other methods. Additionally, T-k-means achieves noticeable gains in accuracy on specific datasets, e.g. Usokin (T-k-means: 0.45, second best (fz-k-means): 0.10), Limb (T-k-means: 0.66, second best (k-means): 0.14), Trachea (T-k-means 0.58, second best (WAC): 0.23), Lung (T-k-means: 0.49, second best (WAC): 0.19) and Diaphragm (T-k-means: 0.83, second best (k-means): 0.25). Results demonstrate that p-ClustVal can greatly boost the clustering accuracy of k-means. Interestingly, the observed gain is conserved across datasets, which further underscores the robustness of p-ClustVal to varying data sizes, number of dimensions and number of clusters.

p-ClustVal enhances classical clustering

The results are reported in Table 2, and highlight the significance of p-ClustVal in enhancing clustering performance, as evidenced by the ARI scores. On the Spleen dataset, although the ARI of EM on raw data surpassed its transformed counterpart, we find that WAC (on transformed data) achieved substantially better ARI scores than EM. When applied to the transformed data, WAC outperformed EM by noticeable margin of 0.36. This shows that for Spleen dataset, better performance can be attained by using WAC on transformed data. The transformation's impact is further emphasized by the maximum ARI improvements observed across datasets. For example, k-means's ARI increased by an impressive 0.58 on the Diaphragm dataset, while Fuzzy-k-means saw a remarkable 0.74 improvement on the Limb dataset. Similarly, the ARI

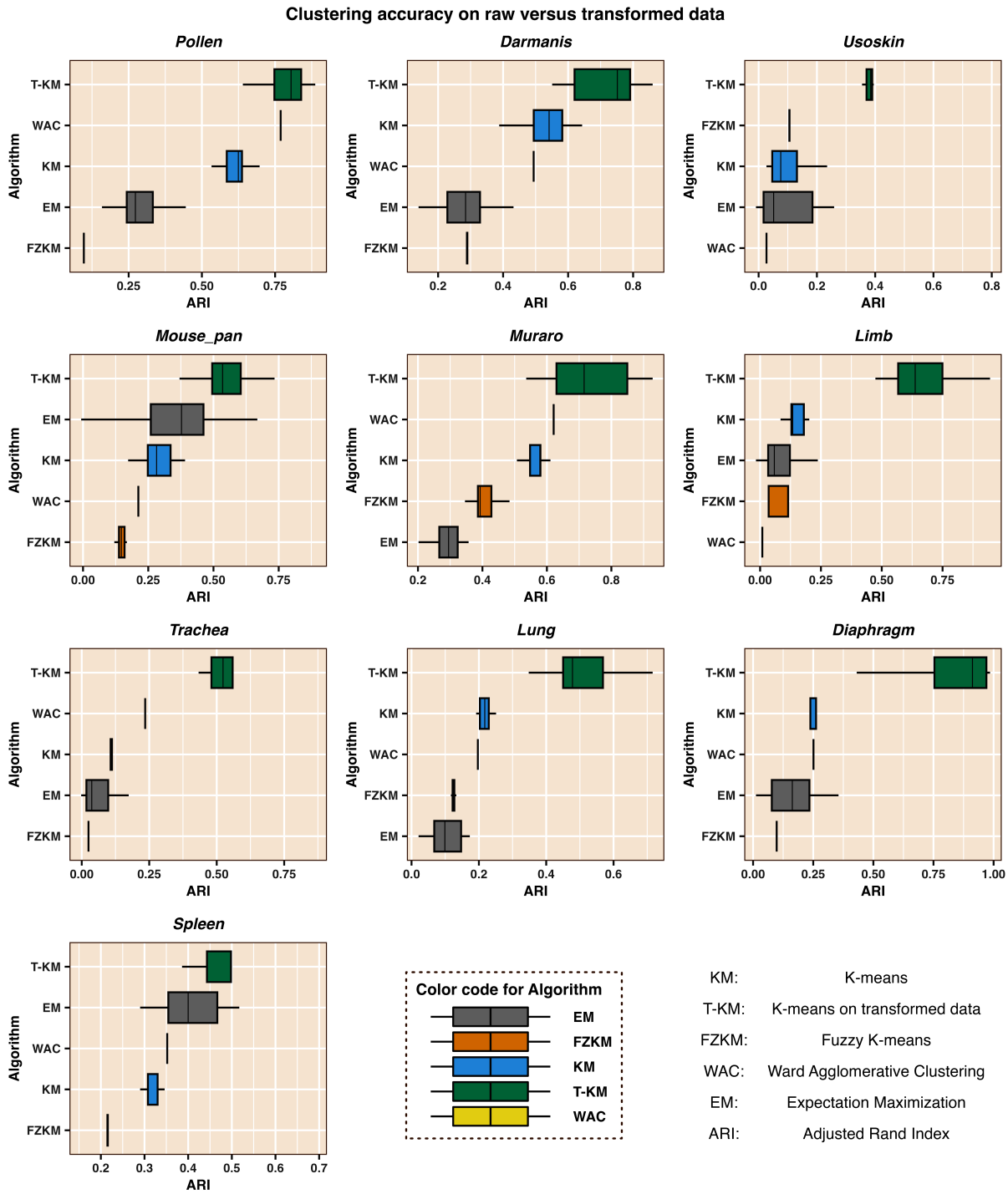

Fig. 6: T-k-means augmented with p-ClustVal performs significantly better, and surpass other algorithms. The gain in accuracy is consistent across datasets and k.

of EM increased by 0.43 on both the Limb and Diaphragm datasets, and WAC's performance was enhanced by 0.72 on the Diaphragm dataset. On majority of datasets, applying the transformation

Table 2: ARI of clustering methods on raw versus transformed data

Dataset	k-means		FZKM		EM		WAC	
	R	T	R	T	R	T	R	T
Pollen	0.60	**0.79**	0.09	**0.35**	0.28	**0.61**	0.76	**0.95**
Darmanis	0.52	**0.72**	**0.27**	0.25	0.24	**0.30**	0.49	**0.65**
Usoskin	0.09	**0.45**	0.10	**0.12**	0.07	**0.47**	0.02	**0.69**
Mouse Pan.	0.29	**0.56**	0.15	**0.54**	0.37	**0.45**	0.21	**0.48**
Muraro	0.56	**0.74**	0.40	**0.79**	0.36	**0.44**	0.62	**0.92**
Limb	0.14	**0.66**	0.05	**0.74**	0.10	**0.53**	0.008	**0.57**
Trachea	0.11	**0.58**	0.02	**0.59**	0.05	**0.43**	0.23	**0.45**
Lung	0.21	**0.49**	0.12	**0.71**	0.11	**0.53**	0.19	**0.42**
Diaphragm	0.25	**0.83**	0.09	**0.55**	0.15	**0.63**	0.25	**0.97**
Spleen	0.31	**0.48**	0.21	**0.27**	**0.41**	0.23	0.35	**0.77**

'R' and 'T' indicate raw and transformed data, respectively. **Bold** indicate superior results. Mouse pan. is Mouse Pancreatic dataset.

consistently improved the ARI of clustering algorithms, demonstrating the complementary nature of p-ClustVal.

p-ClustVal improves clustering with dimension reduction

The empirical evaluation (Table 3) showcases the benefit of p-ClustVal in enhancing the performance of widely-adopted dimensionality reduction techniques. For example, using PCA on transformed data leads to maximum improvement of 0.61 on Limb data. With tSNE, maximum gain of 0.59 is achieved on Usoskin data. When using UMAP, better results are obtained on 8 of 10 datasets, with maximum gain of 0.53 on Usoskin data. While the relative performance of dimensionality reduction methods varied across datasets, the empirical analysis consistently underscored the substantial improvements in downstream clustering quality facilitated by the application of the p-ClustVal. These findings support the application of p-ClustVal as a promising approach for refining dimensionality reduction in the context of high-dimensional genomics data analysis.

p-adic-transformation augments SOTA domain specific clustering

Table 4 show the empirical results. There is reasonable variation in the accuracy of different tools, even when analyzing the same dataset. Notably, no single method unanimously outperform others. However, on most datasets, p-ClustVal either improves the performance significantly or achieve similar performance as raw data. In Seurat, performance is conserved on 6 of 10 datasets. On RaceID, transformed data achieve better accuracy on 7 out of 10 datasets with impressive gain of 0.84 on Diaphragm data, and raw data is only marginally better on Mouse pancreatic and Muraro. In SIMLR, transformed data outperformed the raw data on all 10 datasets with maximum improvement of 0.60 on Lung data. In SC3, accuracy is conserved on 6 of 10 datasets, and in ADClust, transformed data does much better on 9 datasets with dramatic accuracy margin of 0.97 points on Diaphragm data. To summarize, out of 50 independent observations, using transformed data achieved better or similar results in 86% of cases, while raw data is marginally better in 14% cases.

6 Limitation

Finally, we address the limitations of our approach and discuss the possible solutions. As mentioned in Sec. 1, various factors impact the clustering of scRNASeq data (noise, sparsity, biological and technical variation etc.). However, p-ClustVal address the clustering from a data-centric perspective (by preparing an alternate data representation), hence, primary limitation arise due to

Table 3: Comparing the ARI with dimension reduction

Dataset	PCA		tSNE		UMAP	
	R	T	R	T	R	T
Pollen	0.66	**0.74**	0.73	**0.74**	0.64	**0.67**
Darmanis	0.57	**0.74**	0.50	**0.60**	0.44	**0.62**
Usoskin	0.12	**0.55**	0.06	**0.60**	0.04	**0.54**
Mouse pan.	0.21	**0.48**	0.31	**0.36**	0.36	**0.40**
Muraro	0.51	**0.83**	0.63	0.62	0.63	**0.76**
Limb	0.08	**0.62**	0.35	**0.51**	0.34	**0.51**
Trachea	0.12	**0.55**	0.31	**0.55**	**0.72**	0.51
Lung	0.20	**0.40**	30	**0.33**	0.31	**0.37**
Diaphragm	0.25	**0.81**	0.38	**0.53**	0.40	**0.62**
Spleen	0.30	**0.52**	0.25	0.25	**0.31**	0.28

'R' and 'T' indicate raw and transformed data, respectively. **Bold** indicate better results and blue indicate either similar performance or when difference is less than 0.01.

data-driven parameter estimate. Specifically, p-ClustVal is influenced by the accuracy of the `coh` and `sep` parameters, as these parameters directly impact the scaling and relative discretization of data. As described in Sec. 3, `coh` is determined by identifying the neighborhood of each data point. This approach inherently assumes that a data point's neighborhood primarily consists of data from the same cluster, implying homogeneity. However, this assumption may not hold in datasets with very high variance, where different clusters may frequently overlap, resulting in heterogeneous neighborhoods. This heterogeneity can lead to inaccurate estimates of `coh`, as the top k neighbors might include data points from other clusters, skewing the estimate towards higher variance data.

For example, on Adam data with higher overlap between classes, p-ClustVal showed marginal improvement over raw data (Table 5). Recently, feature extraction methods, which classify points in an alternate, lower-dimensional space, have been shown to enhance clustering [73, 74]. Consequently, we show that it's possible to recover the performance of p-ClustVal by doing parameters search in a low-dimensional space where neighborhood is relatively homogeneous. It's important to note that the homogeneity assumption is a reasonable simplification of the problem, allowing for a data-centric, user-independent search for `coh` and `sep` based solely on the data. Moreover, empirical validation has shown that the assumption is effective on majority of datasets considered in this study.

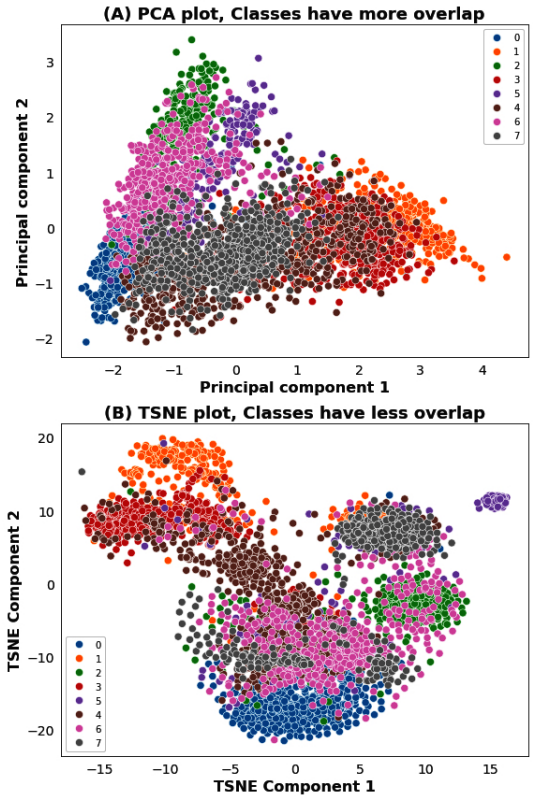

Fig. 7: Visualizing the Adam data. Compared to PCA, TSNE reduces the relative overlap of classes.

To better illustrate this, Fig. 7-A visualizes the Adam dataset on the top two principal components, with data colored according to true class labels. The high degree of overlap between different clusters is evident. A corresponding t-SNE visualization (Fig. 7-B) show that clusters are more distinct after t-SNE, making it a better candidate for parameter selection. We conducted a neighbor search in the default data space, PCA and t-SNE space, followed by 20 trials of k-means. The results, presented in Table 5, indicate that the default strategy only marginally improves over raw data, whereas parameters derived from PCA and t-SNE based neighbor searches yield reasonable improvement. These finding opens avenues for further research into automated methods that

Table 4: Comparing state of the art algorithms on raw and transformed scRNASeq data

Dataset	Adjusted Rand Index (ARI)									
	Seurat		RaceID		SIMLR		SC3		ADClust	
	R	T	R	T	R	T	R	T	R	T
Pollen	0.77	**0.8**	0.58	**0.82**	0.09	**0.5**	0.97	0.97	0.10	**0.68**
Darmanis	0.61	**0.66**	0.46	**0.51**	0.59	**0.84**	**0.95**	0.84	0.23	**0.84**
Usoskin	**0.64**	0.57	NA	NA	NA	NA	0.88	0.88	0.18	**0.27**
Mouse pan.	0.46	0.46	**0.31**	0.25	0.29	**0.47**	**0.41**	0.36	0.54	**0.58**
Muraro	0.93	0.93	**0.83**	0.78	NA	NA	0.94	0.93	0.39	**0.93**
Limb	0.66	0.67	0.5	**0.56**	0.24	**0.52**	0.64	**0.94**	0.04	**0.58**
Trachea	**0.95**	0.93	0.49	**0.54**	0.31	**0.51**	0.48	0.49	0.18	**0.90**
Lung	0.6	0.6	0.17	**0.41**	0.16	**0.76**	0.48	0.49	0.16	**0.59**
Diaphragm	0.98	0.98	0.13	**0.97**	0.31	**0.65**	0.99	0.99	0.02	**0.98**
Spleen	0.89	0.88	0.40	**0.44**	0.22	**0.5**	NA	NA	**0.63**	0.30

'R' and 'T' indicate raw and transformed data, respectively. **Bold** indicate better results and blue indicate either similar performance or when difference is less than 0.01. NA indicate method failed to run on specific data.

leverage such representations to enhance parameter search and clustering performance.

Table 5: Adam dataset. m: 3660, d: 12976, Sparsity: 86.07%

Dataset	Default		PCA	tSNE
	R	T	T	T
Adam	0.02	0.09	0.21	**0.33**

'R' and 'T' indicate raw and transformed data. **Bold** indicate better results.

7 Summary & future work

In this work, we introduced p-ClustVal, an innovative approach towards creating alternate representations of high dimensional genomics data. Drawing inspiration from the mathematical theory of p-adic numbers, p-ClustVal leverages metric spaces based on p-adic-valuation to enhance cluster separation and discernibility. p-ClustVal combined with an unsupervised heuristic for parameter estimation, is validated on downstream clustering, and dimension reduction methods, minimizes tuning requirements and shows remarkable adaptability across various datasets. Comprehensive evaluation on SOTA domain-specific tools demonstrates the robustness of this approach across datasets, and diverse clustering techniques. While p-ClustVal currently faces challenges with high cluster overlap, our future direction involves integrating more sophisticated low dimensional embedding to refine data representations, aiming to overcome these limitations. p-ClustVal is a novel example of applying abstract mathematical concepts for solving complex real-world problems, and opens up new avenues for interdisciplinary research in mathematics, computer science, and biology.

Acknowledgements. Authors thank the technical support extended by the system administration team of Luddy School of Informatics, Computing & Engineering for their help in deploying the system infrastructure for this study.

Author Contributions. PS: Conceptualization, Methodology, Development & Design, Experiment, Investigation, Writing – original draft, Writing – review & editing, Visualization. SM: Methodology, Development & Design, Validation, Investigation, Writing – original draft, Writing – review & editing. HK: Methodology, Investigation, Supervision, Writing – original draft, Writing – review & editing. MD: Writing – original draft, Writing – review & editing, Resources, Supervision.

Funding. This work is partially supported by the funding from National Science Foundation grant DBI-2146866.

Data availability. Relevant data sources are cited in the manuscript.

Declarations

- **Conflict of interest** The authors **do not** declare any conflict of interests.
- **Ethics approval** Not applicable.
- **Consent to participate** Not applicable.
- **Consent for publication** All authors gave their consent for the publication.

References

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8*, volume 1, pages 420–434. Springer, 2001.

[2] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann. The technology and biology of single-cell rna sequencing. *Molecular cell*, 58(4):610–620, 2015.

[3] Efthymia Papalexi and Rahul Satija. Single-cell rna sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology*, 18(1):35–45, 2018.

[4] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

[5] Dominic Grün, Anna Lyubimova, Lennart Kester, Kay Wiebrands, Onur Basak, Nobuo Sasaki, Hans Clevers, and Alexander Van Oudenaarden. Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature*, 525(7568):251–255, 2015.

[6] Alessandra Dal Molin, Giacomo Baruzzo, and Barbara Di Camillo. Single-cell rna-sequencing: assessment of differential expression analysis methods. *Frontiers in genetics*, 8:62, 2017.

[7] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8): 651–666, 2010.

[8] Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3): 133–145, 2015.

[9] Aaron TL Lun, Davis J McCarthy, and John C Marioni. A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. *F1000Research*, 5, 2016.

[10] Kurt Hensel. Über eine neue begründung der theorie der algebraischen zahlen. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 6:83–88, 1897.

[11] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21:1–35, 2020.

[12] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[13] Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature communications*, 10(1):5416, 2019.

[14] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[15] Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. Using neural networks for reducing the dimensions of single-cell rna-seq data. *Nucleic acids research*, 45(17): e156–e156, 2017.

[16] Xiangyu Li, Weizheng Chen, Yang Chen, Xuegong Zhang, Jin Gu, and Michael Q Zhang. Network embedding-based representation learning for single cell rna-seq data. *Nucleic acids research*, 45(19):e166–e166, 2017.

[17] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[18] 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.

[19] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[20] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

[21] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 29, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015408. URL https://doi.org/10.1145/1015330.1015408.

[22] Charles Elkan. Using the triangle inequality to accelerate k-means. In *Proceedings of the 20th international conference on Machine Learning (ICML-03)*, pages 147–153, 2003.

[23] Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483–486, 2017.

[24] Justina Žurauskienė and Christopher Yau. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics*, 17:1–11, 2016.

[25] Lu Yang, Jiancheng Liu, Qiang Lu, Arthur D Riggs, and Xiwei Wu. Saic: an iterative clustering approach for analysis of single cell rna-seq data. *BMC genomics*, 18:9–17, 2017.

[26] Eugenio Marco, Robert L Karp, Guoji Guo, Paul Robson, Adam H Hart, Lorenzo Trippa, and Guo-Cheng Yuan. Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proceedings of the National Academy of Sciences*, 111(52):E5643–E5650, 2014.

[27] Dominic Grün, Mauro J Muraro, Jean-Charles Boisset, Kay Wiebrands, Anna Lyubimova, Gitanjali Dharmadhikari, Maaike van den Born, Johan Van Es, Erik Jansen, Hans Clevers, et al. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell stem cell*, 19(2):266–277, 2016.

[28] LKPJ Rdusseeun and P Kaufman. Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*, volume 31, 1987.

[29] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[30] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.

[31] Peijie Lin, Michael Troup, and Joshua WK Ho. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome biology*, 18(1):1–11, 2017.

[32] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[33] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

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

[35] Bo Wang, Daniele Ramazzotti, Luca De Sano, Junjie Zhu, Emma Pierson, and Serafim Batzoglou. Simlr: A tool for large-scale genomic analyses by multi-kernel learning. *Proteomics*, 18(2):1700232, 2018.

[36] Tian Tian, Ji Wan, Qi Song, and Zhi Wei. Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4):191–198, 2019.

[37] Liang Chen, Weinan Wang, Yuyao Zhai, and Minghua Deng. Deep soft k-means clustering with self-training for single-cell rna sequence data. *NAR genomics and bioinformatics*, 2 (2):lqaa039, 2020.

[38] Yuansong Zeng, Zhuoyi Wei, Fengqi Zhong, Zixiang Pan, Yutong Lu, and Yuedong Yang. A parameter-free deep embedded clustering method for single-cell rna-seq data. *Briefings in Bioinformatics*, 23(5):bbac172, 2022.

[39] Yuchen Yang, Ruth Huh, Houston W Culpepper, Yuan Lin, Michael I Love, and Yun Li. Safe-clustering: single-cell aggregated (from ensemble) clustering for single-cell rna-seq data. *Bioinformatics*, 35(8):1269–1277, 2019.

[40] Xiaoshu Zhu, Jian Li, Hong-Dong Li, Miao Xie, and Jianxin Wang. Sc-gpe: a graph partitioning-based cluster ensemble method for single-cell. *Frontiers in Genetics*, 11: 604790, 2020.

[41] Yixiang Huang, Hao Jiang, and Wai-Ki Ching. scewe: high-order element-wise weighted ensemble clustering for heterogeneity analysis of single-cell rna-sequencing data. *Briefings in Bioinformatics*, 25(3):bbae203, 2024.

[42] Wenming Wu, Zaiyi Liu, and Xiaoke Ma. jsrc: a flexible and accurate joint learning algorithm for clustering of single-cell rna-sequencing data. *Briefings in bioinformatics*, 22(5):bbaa433, 2021.

[43] Qi Yang, Zhaochun Xu, Wenyang Zhou, Pingping Wang, Qinghua Jiang, and Liran Juan. Aninterpretable single-cell rna sequencing data clustering method based on latent dirichlet allocation. *Briefings in Bioinformatics*, 24(4):bbad199, 2023.

[44] Yaxuan Cui, Shaoqiang Zhang, Ying Liang, Xiangyun Wang, Thomas N Ferraro, and Yong Chen. Consensus clustering of single-cell rna-seq data by enhancing network affinity. *Briefings in bioinformatics*, 22(6): bbab236, 2021.

[45] Edwin Vans, Ashwini Patil, and Alok Sharma. Feats: feature selection-based clustering of single-cell rna-seq data. *Briefings in Bioinformatics*, 22(4):bbaa306, 2021.

[46] Jing Wang, Junfeng Xia, Dayu Tan, Rongxin Lin, Yansen Su, and Chun-Hou Zheng. schfc: a hybrid fuzzy clustering method for single-cell rna-seq data optimized by natural computation. *Briefings in Bioinformatics*, 23(2): bbab588, 2022.

[47] Hao Hua and Ludger Hovestadt. p-adic numbers encode complex networks. *Scientific Reports*, 11(1):17, 2021.

[48] Branko Dragovich and Nataša Ž Mišić. p-adic hierarchical properties of the genetic code. *Biosystems*, 185:104017, 2019.

[49] Wilson A Zuniga-Galindo. p-adic statistical field theory and deep belief networks. *Physica A: Statistical Mechanics and its Applications*, 612:128492, 2023.

[50] BA Zambrano-Luna and WA Zúñiga-Galindo. p-adic cellular neural networks: Applications to image processing. *Physica D: Nonlinear Phenomena*, 446:133668, 2023.

[51] Fernando Q Gouvea. A marvelous proof. *The American Mathematical Monthly*, 101 (3):203–222, 1994.

[52] Fionn Murtagh. Sparse p-adic data coding for computationally efficient and effective big data analytics. *P-Adic Numbers, Ultrametric Analysis, and Applications*, 8:236–247, 2016.

[53] Jenny Benois-Pineau, Andrei Khrennikov, and Nikolai Kotovitch. Image segmentation in compressed domain by clustering methods with euclidean and p-adic metrics. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 969–976, 2002.

[54] Elif Kartal, Fatma Çalışkan, Beyaz Başak Eskişehirli, and Zeki Özen. p-adic distance and k-nearest neighbor classification. *Neurocomputing*, 578:127400, 2024.

[55] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

[56] Tallulah S Andrews and Martin Hemberg. Identifying cell populations with scrnaseq. *Molecular aspects of medicine*, 59:114–122, 2018.

[57] Angelo Duò, Mark D Robinson, and Charlotte Soneson. A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research*, 7, 2018.

[58] Fernando Q Gouvêa and Fernando Q Gouvêa. *p-adic Numbers*. Springer, 1997.

[59] Angela R Wu, Norma F Neff, Tomer Kalisky, Piero Dalerba, Barbara Treutlein, Michael E Rothenberg, Francis M Mburu, Gary L Mantalas, Sopheak Sim, Michael F Clarke, et al.

Quantitative assessment of single-cell rna-sequencing methods. *Nature methods*, 11(1): 41–46, 2014.

[60] Giuseppe Ciaburro, V Kishore Ayyadevara, and Alexis Perrier. *Hands-on machine learning on google cloud platform: Implementing smart and efficient analytics using cloud ml engine*. Packt Publishing Ltd, 2018.

[61] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1): 1–15, 2017.

[62] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

[63] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4): 824–836, 2018.

[64] Vincent Garcia, Eric Debreuve, and Michel Barlaud. Fast k nearest neighbor search using gpu. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6. IEEE, 2008.

[65] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature biotechnology*, 32(10):1053–1058, 2014.

[66] Spyros Darmanis, Steven A Sloan, Derek Croote, Marco Mignardi, Sophia Chernikova, Peyman Samghababi, Ye Zhang, Norma Neff, Mark Kowarsky, Christine Caneda, et al. Single-cell rna-seq analysis of infiltrating neoplastic cells at the migrating front of human glioblastoma. *Cell reports*, 21(5):1399–1410, 2017.

[67] Dmitry Usoskin, Alessandro Furlan, Saiful Islam, Hind Abdo, Peter Lönnerberg, Daohua Lou, Jens Hjerling-Leffler, Jesper Haeggström, Olga Kharchenko, Peter V Kharchenko, et al. Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing. *Nature neuroscience*, 18(1): 145–153, 2015.

[68] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

[69] Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon Van Gurp, Marten A Engelse, Francoise Carlotti, Eelco Jp De Koning, et al. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4): 385–394, 2016.

[70] Nicholas Schaum, Jim Karkanias, Norma F Neff, Andrew P May, Stephen R Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B Chen, et al. Single-cell transcriptomics of 20 mouse organs creates a tabula muris: The tabula muris consortium. *Nature*, 562(7727):367, 2018.

[71] Yanglan Gan, Ning Li, Guobing Zou, Yongchang Xin, and Jihong Guan. Identification of cancer subtypes from single-cell rna-seq data using a consensus clustering method. *BMC medical genomics*, 11:65–72, 2018.

[72] Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, and Serafim Batzoglou. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nature methods*, 14(4):414–416, 2017.

[73] Honghua Liu, Jing Yang, Ming Ye, Scott C James, Zhonghua Tang, Jie Dong, and Tongju Xing. Using t-distributed stochastic neighbor embedding (t-sne) for cluster analysis and spatial zone delineation of groundwater geochemistry data. *Journal of Hydrology*, 597:126146, 2021.

[74] Farzana Anowar, Samira Sadaoui, and Bassant Selim. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review*, 40: 100378, 2021.

[75] Teeratorn Kadeethum, Daniel O'Malley, Jan Niklas Fuhg, Youngsoo Choi, Jonghyun

Lee, Hari S Viswanathan, and Nikolaos Bouklas. A framework for data-driven solution and parameter estimation of pdes using conditional generative adversarial networks. *Nature Computational Science*, 1(12):819–829, 2021.

[76] S Ashwin Renganathan, Romit Maulik, Stefano Letizia, and Giacomo Valerio Iungo. Data-driven wind turbine wake modeling via probabilistic machine learning. *Neural Computing and Applications*, pages 1–16, 2022.

[77] Massimo Andreatta, Ariel J Berenstein, and Santiago J Carmona. scgate: marker-based purification of cell types from heterogeneous single-cell rna-seq datasets. *Bioinformatics*, 38(9):2642–2644, 2022.

[78] Tian Tian, Cheng Zhong, Xiang Lin, Zhi Wei, and Hakon Hakonarson. Complex hierarchical structures in single-cell genomics data unveiled by deep hyperbolic manifold learning. *Genome Research*, 33(2):232–246, 2023.

[79] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

Appendix A Determining number of neighbors (k)

Table A1 reports the ARI on different values of k (neighbors) for 5 different datasets. The ARI is in similar range despite changes in k, and we opted for $k = 5$ in our experiments.

Table A1: Comparing ARI on different number of neighbors

	Adjusted Rand Index (ARI)					
Data	$k = 2$	$k = 5$	$k = 10$	$k = 15$	$k = 20$	$k = 25$
Pollen	0.861	0.856	0.870	0.860	0.879	0.859
Usoskin	0.668	0.696	0.665	0.641	0.671	0.677
Mouse Pancreatic	0.461	0.477	0.511	0.479	0.479	0.460
Trachea	0.550	0.552	0.550	0.550	0.551	0.553
Spleen	0.618	0.579	0.555	0.611	0.586	0.620

Appendix B Motivation for Data-Centric Parameter Search

The data-driven parameters have been shown to improve methods in learning [75, 76]. [76] argues that data-driven parameters with deep autoencoders and Gaussian process modeling can accurately approximate the wind-turbine wake flow field. [75] has used conditional generative adversarial networks (cGANs) to find solutions to partial differential equations (PDEs) and thus estimate the parameters of the model. They have demonstrated that data-centric parameter estimation can improve accuracy as well as speed. Thus, we also provide a transformation that is tuned with parameters driven by data. The main benefit of data-centric parameter estimation is that it reflects the data more accurately. Compared to the generalized approach where all parameters are one size fits all, this provides more flexibility in tuning.

In our work, we created a transformation function that leverages neighborhood information to infer separation and cohesion parameters. These parameters are then used to discretize the space for the dataset. The approach is unique to each dataset. Contrary to other modeling approaches, which rely on an understanding of the system, data-centric parameter approaches learn directly from the data, providing more freedom. These approaches can be useful in systems such as single-cell RNA-seq, where the underlying system is very complex, making modeling very difficult [77, 78]. Empirically, we have shown improvements in different classes of clustering methods using our transformation.

Appendix C Data processing

For traditional clustering and dimension reduction, standard data processing was done, for example: genes expressed in less than 1% cells are removed from data followed by removal of cells that have less than 100 genes expressed (non-zero count in less than 100 genes). For data normalization (adjusting for differences in gene counts between cells), we use the method proposed in [79] which ensure that total gene counts in all cells is the same-this removes the bias in cell size (total gene counts of cells) due to sequencing technology or library preparation steps. Finally, p-ClustVal is applied on the normalized data.

For experiments with scRNA-seq packages, in Seurat, cells that have non-zero count in more than 100 genes are kept and genes expressed in less than 3 cells are removed. Normalization is done on filtered data by dividing the gene count of each cell by total gene count of that cell and multiplying by the scale factor of 10^4. The normalized data (D) is log transformed as $log(D + 1)$. For SIMLR and RaceID, data is processed by the default pipeline of the package. SC3 and ADClust does not perform explicit data filtering, so data is processed similar to traditional clustering experiments.

Appendix D Computing Platform

Experiments on real and synthetic data were done on a 64-bit Ubuntu Linux system with 1.5 TB of memory and 24 dual socket Intel Xeon cores.