A monotonicity-based gene clustering algorithm for enhancing clarity in single-cell RNA sequencing data

Victor Wang
victorwang37@gmail.com
University of Houston

Pietro Antonio Cicalese
pcicalese@uh.edu
University of Houston

Chandra Mohan
cmohan@central.uh.edu
University of Houston

Abstract

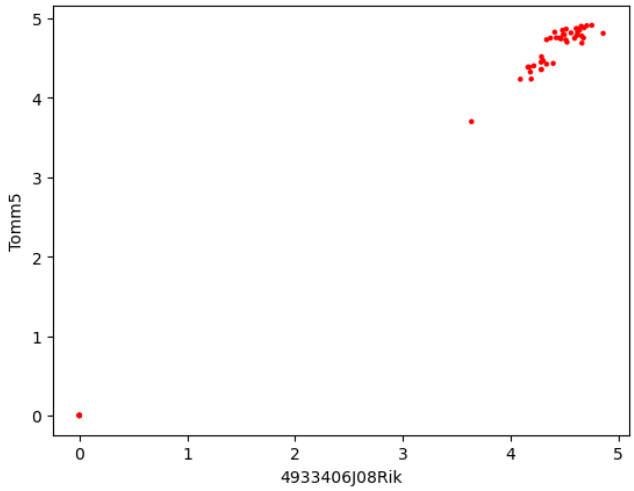
Single-cell RNA sequencing (scRNA-seq) technologies and analysis tools have allowed for meaningful insight into the roles and relationships of cells. However, high dimensionality, frequent dropout values, and technical noise remain prevalent challenges for scRNA-seq data, obscuring the already complex expression patterns. To address several shortcomings in commonly used distance metrics, we present a monotonicity-based distance metric designed to enhance the clarity of scRNA-seq data. We apply our metric in a gene clustering algorithm, which we run on several biological datasets. We compare our results to those generated by popular clustering algorithms to demonstrate that our algorithm has substantial ability to improve the accuracy of subsequent cell clustering.

1 Background

A scRNA-seq dataset consists of an expression matrix of genes by cells, measuring the degree to which each gene is expressed in each cell. Many computational methods have been developed for using scRNA-seq data to identify subpopulations of cells [1, 2, 3]. Due to the high dimensionality of scRNA-seq data and the predominance of dropout values, the data tends to be sparse, causing unsupervised cell clustering algorithms, which compute and use the distances between cells, to be naturally susceptible to the curse of dimensionality [4, 10]. As a result, similarities between cells may be difficult to interpret. Given that the expression values of thousands of genes are measured in most scRNA-seq datasets, many co-expressed genes are likely to be present, rendering cell-to-cell distances even more difficult to interpret [6]. Therefore, we hypothesized that pre-processing scRNA-seq data by clustering together co-expressed genes has the potential to organize data for easier interpretability by cell clustering algorithms.

Gene clustering algorithms can be classified as supervised or unsupervised. Supervised algorithms usually aim to cluster genes with a shared functional category [7], whereas unsupervised algorithms cluster genes based on expression data without a priori information [6]. The lack of readily available functional annotations for all genes and the sensitivity of scRNA-seq data to the nuances of varying biological settings suggest that more individualized clusters are obtainable from unsupervised approaches [8], which are what this paper will limit itself to discussing.

The high dimensionality and abundant technical noise present in scRNA-seq data make gene expression patterns difficult to accurately model. Many distance metrics have been explored, including Pearson, Spearman, and Euclidean [2, 6, 9]. Pre-processing procedures, such as quality control, log-transformation, and normalization, have been developed to allow data to be interpreted more easily by clustering algorithms [22]. However, discovering ways to cope with technical noise is a field of ongoing exploration [10], suggesting that gene expression patterns may be too complex and noisy to be effectively modeled by strictly linear correlation (i.e., Pearson correlation) in spite of popular pre-processing methods. This obstacle motivates a distance metric that quantifies the monotonicity (i.e., the tendency for one variable to increase with another, not necessarily linearly) of the relationship between the expression values of a pair of genes. Spearman's correlation and Kendall's correlation are metrics that operate on the relative ranks of values in an effort to quantify monotonicity [14]. However, one shortcoming shared by rank-based correlations is that in the process of compressing values into their ranks, information derivable from the relative magnitudes of the values is lost.



To illustrate, for the graph to the left (which plots the pre-processed expression values of two genes taken from the Biase dataset [24]), the Pearson coefficient is a strong 0.998, and although rank-based monotonicity correlations are designed to be lenient in recognizing non-linear association, the Spearman coefficient is 0.898, and the Kendall tau-b coefficient is only 0.753. For each coefficient, 1 indicates perfect association, and 0 indicates random association. Coefficients were obtained using the scipy Python package [18]. This relationship exhibits a clear tendency for the x coordinate to increase with the y coordinate, but the rank-based correlations are misled into over-magnifying the relatively minor distances among the tightly packed points.

To address the shortcomings in the distance metrics discussed above, we present a monotonicity-based distance metric that accounts for the magnitudes of the distances between points. To demonstrate the effectiveness of our distance metric when applied in a gene clustering algorithm, we compare our algorithm to four other methods (k-means, hierarchical clustering, principal component analysis, and Clust, all of which have yielded strong results in previous work [15, 16, 23, 8]) by running the resulting clustered matrices through SC3, a cell clustering procedure [2].

2 Results

Summary of Datasets

Dataset	k	n	m	Units
Biase	3	49	6323	FPKM
Shen	3	1735	5864	CPM
Goolam	5	124	11157	CPM
Yan	8	124	9460	RPKM
Deng	10	259	10372	RPKM
Pollen	11	301	11828	TPM
Tatsuoka	2	24	5255	RPKM
Bozec	3	1391	6203	CPM
Treutlein	4	201	9462	FPKM
Patel	5	430	5948	TPM

k is the number of cell clusters, n is the number of cells analyzed, and m is the number of genes after pre-processing (see Methods).

FPKM means fragments per kilobase of transcript per million mapped reads, CPM means counts per million mapped reads, RPKM means reads per kilobase of transcript per million mapped reads, and TPM means transcripts per million mapped reads.

In each dataset, we chose to use cell labels that represent distinct biological states or conditions [24-33]. Therefore, the cell labels are highly objective.

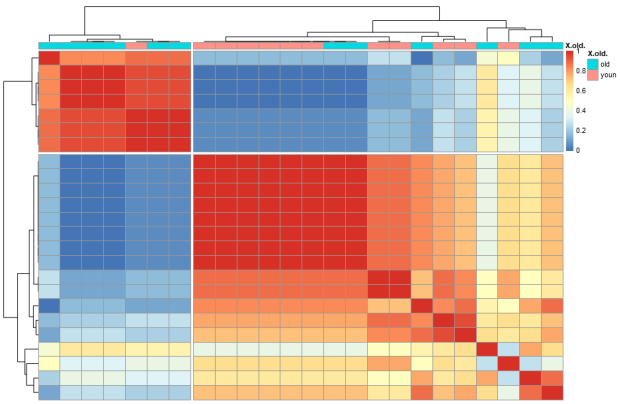
We used each dataset as follows. We first ran SC3 on the unclustered (pre-processed) matrix to generate a benchmark cell clustering. For each of the 6 gene clustering algorithms that we tested (ours, k-means, agglomerative hierarchical clustering using Euclidean distance (HCE), agglomerative hierarchical clustering using Pearson correlation distance (HCP), principal component analysis (PCA), and Clust), we used the algorithm to produce a compressed feature matrix which we subsequently ran through SC3. We used the Adjusted Rand Index (ARI) to evaluate how closely the resulting cell clustering matched the one provided by the publishers of the datasets (see Methods). A higher ARI indicates a closer match, with an ARI of 1 indicating exact correspondence and an ARI of 0 indicating random correspondence. Although SC3 has been shown to produce more stable results than other popular cell clustering algorithms, its performance still experiences fluctuation from run to run [2]. Therefore, we repeated each trial 10 times. Our results are summarized in the table below. Red or green text indicates a statistically significant ARI decrease or increase, respectively (one-sided Wilcoxon signed-rank test p-value < 0.01 [11]). Bolded text indicates a median ARI that differs from the median ARI without gene clustering by 0.1 or more.

Table 1

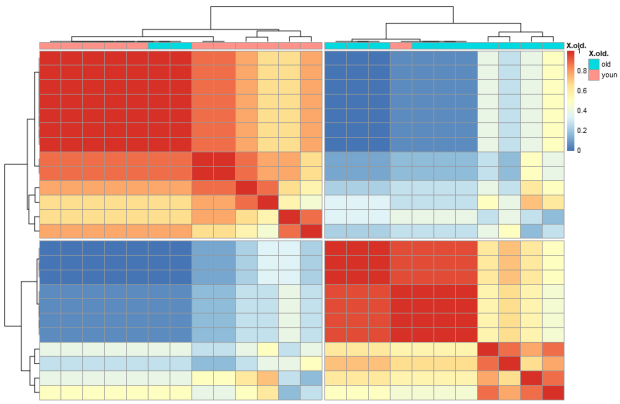
Method / Dataset	No gene clustering	Ours	K-means	Hierarchical (Euclidean)	Hierarchical (Pearson)	PCA	Clust	
Biase	6323	2400	2400	2400	2400	48	6244	# features
	0.948	0.948	0.948	0.948	**0.359**	0.948	0.948	Median ARI
	0:18	13:48	6:42	0:03	0:03	0:00	0:39	CPU time
Shen	5864	5863	5863	5863	5863	500	5864	# features
	0.346	0.345	0.346	0.346	0.345	**0.692**	0.346	Median ARI
	51:08	7:05:06	3:45:04	0:30	0:32	12:40	7:43	CPU time
Goolam	11157	10111	10111	10111	10111	123	11157	# features
	0.687	0.687	0.687	0.687	0.687	0.687	0.687	Median ARI
	0:21	1:40:34	1:22:26	0:09	0:09	0:01	4:20	CPU time
Yan	9460	5515	5515	5515	5515	123	9460	# features
	0.795	0.767	0.786	0.759	0.715	0.732	0.795	Median ARI
	0:22	1:16:48	43:21	0:08	0:07	0:01	1:06	CPU time
Deng	10372	8181	8181	8181	8181	258	10372	# features
	0.553	0.604	0.615	0.620	0.511	0.543	0.553	Median ARI
	0:44	3:55:51	1:48:10	0:18	0:15	0:03	8:03	CPU time
Pollen	11828	10658	10658	10658	10658	300	11828	# features
	0.960	0.965	0.960	0.960	0.966	**0.851**	0.960%	Median ARI
	0:53	5:16:04	3:27:45	0:21	0:20	0:07	10:46	CPU time
Tatsuoka	5255	2050	2050	2050	2050	23	5229	# features
	0.143	**0.543**	**0.420**	**0.420**	0.220	0.138	0.143	Median ARI
	0:18	3:47	5:02	0:03	0:02	0:00	0:30	CPU time
Bozec	6203	6165	6165	6165	6165	500	6203	# features
	0.306	0.307	0.295	0.307	0.281	0.289	0.306	Median ARI
	18:47	5:50:12	3:08:45	0:27	0:28	9:05	6:36	CPU time
Treutlein	9462	8711	8711	8711	8711	200	9462	# features
	0.647	**0.808**	0.647	0.647	0.637	**0.761**	0.647	Median ARI
	0:32	2:04:45	1:27:57	0:12	0:13	0:03	1:14	CPU time
Patel	5948	5856	5856	5856	5856	429	5881	# features
	0.994	0.983	0.994	0.989	0.994	0.979	0.984	Median ARI
	1:52	3:01:11	1:02:34	0:11	0:18	20:39	1:36	CPU time

CPU times were measured using the Linux `time` command or the `system.time` function in R. In the "No gene clustering" column, we recorded how long SC3 took to run on the unclustered data (the SC3 runtime is excluded from the times recorded in the other columns). Since the `system.time` command is unable to accurately measure the CPU time taken by SC3, the elapsed time was recorded instead (with the number of CPU cores used in SC3 set to 1) in the "No gene clustering" column. See the Methods section for details on how these algorithms were run.

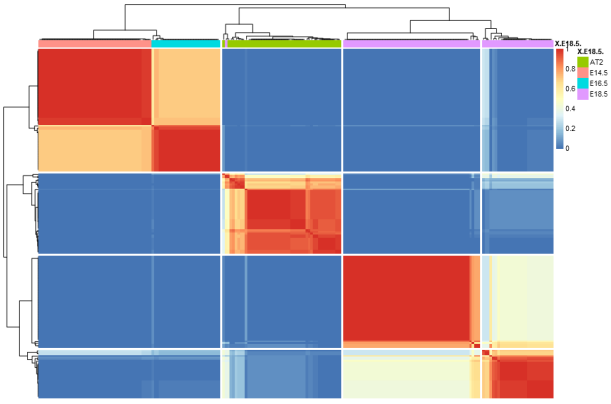
To visualize the effects that our algorithm has on cell clustering, we plot the heatmaps produced by SC3 before and after our clustering, at their respective medians (5th highest ARI), for the Tatsuoka and Treutlein datasets. The SC3 workflow consolidates the clustering results from many combinations of clustering parameters to generate an $n \times n$ consensus matrix, visualized as a heatmap. The colors in the heatmap range from dark blue, indicating high confidence that the two cells belong in different clusters, to dark red, indicating high confidence that the two cells belong in the same cluster. The $n \times n$ matrix is divided into a $k \times k$ block matrix, with the generated clusters occurring along the descending diagonal. In the ideal clustering, the k blocks along the descending diagonal would be completely red and everything else would be completely blue. The cells labels reported by the publishers of the data are color coded above the heatmaps.



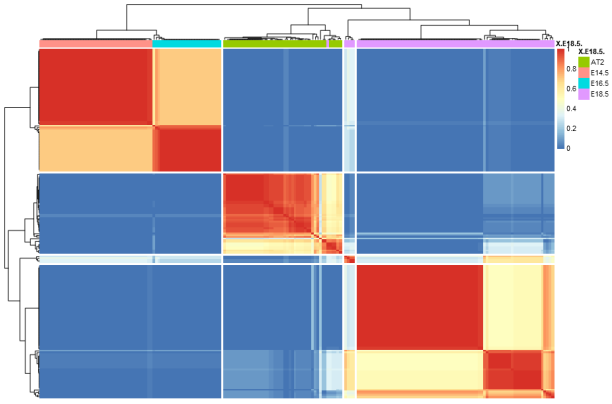
On the unclustered Tatsuoka dataset, 7 out of the 24 cells are misassigned.



On our compressed feature matrix, only 3 cells are misassigned.



On the unclustered Treutlein dataset, the E18.5 cells are split into 2 approximately equally sized clusters.



With our compressed feature matrix, the cell relationships among the E18.5 cells are clarified, leading nearly all the E18.5 cells to be clustered together.

3 Discussion

Identifying cell subpopulations is an important application of scRNA-seq data and has led to notable developments such as the Human Cell Atlas in recent years [6, 21]. We use the ARI to quantitatively compare the accuracy of cell clusterings. The gene clustering algorithms all had minimal effects on the ARI in most of the datasets. In some cases, the gene clustering algorithm considerably decreased the ARI (HCP in the Biase dataset and PCA in the Pollen dataset), suggesting that scRNA-seq data can be sensitive to feature extraction. However, for both of these datasets, the other clustering algorithms were able to maintain the ARI, indicating that feature extraction with minimal information loss was achievable. Therefore, we established that being able to identify the extent to which clustering should be performed is a crucial attribute of an effective clustering algorithm. In other words, recognizing when meaningful clusters are rare is important in avoiding information loss in cases where feature extraction would disrupt data interpretability. Among the algorithms we tested, our algorithm, PCA, and Clust have adjustable parameters but are still able to produce strong results by using default parameters and rules of thumb without extensive intervention. In fact, Clust's consensus-based workflow renders it arguably too reluctant to cluster genes together, which led Clust to hardly reduce, if at all, the number of features in each dataset. This understanding reveals a valuable advantage of our clustering algorithm (as well as PCA and Clust) over k-means and hierarchical clustering. Our clustering algorithm does not rely on a pre-defined number of clusters, making it more applicable to the abundant cases in high dimensional space where the number of clusters is not known beforehand. For example, our algorithm more than halved the number of features in the Tatsuoka dataset but only reduced the number of features in the Treutlein dataset by less than 10%.

In practice, our algorithm would not have been there to supply k-means and hierarchical clustering with a baseline number of clusters to generate, so our results compared to those of k-means and hierarchical clustering are more pessimistic than reality.

Although our algorithm minimally affected most of the datasets, its drastic improvements in the Tatsuoka and Treutlein datasets (mean and median ARI increased by more than 0.1) demonstrate its substantial ability to improve clarity in scRNA-seq data for cell clustering. Our results show that other clustering algorithms are also capable of clarifying cell-to-cell relationships through gene clustering (k-means and HCE in the Tatsuoka dataset, and PCA in the Shen and Treutlein datasets). Therefore, we present our distance metric and clustering algorithm as a strong option to be explored by future researchers, potentially in a consensus clustering approach. Furthermore, as the algorithms we present are not exclusively applicable to scRNA-seq data, we believe our algorithms can find applications in a wider variety of fields.

3.1 Limitations

In this study, although we chose the clustering algorithms that we considered most relevant to scRNA-seq data, the breadth of the clustering techniques we tested could only be so wide. In addition, our algorithm comes at a high computational cost compared to SC3 and the other clustering algorithms we compared (see Methods for the derivation of the time complexity). Therefore, a limitation of our algorithm is lack of scalability to large datasets, especially when the number of genes grows large. A large factor behind the disparity between the runtimes of our algorithm and SC3 is that the number of genes greatly exceeds the number of cells in the datasets we tested. As the number of cells approaches the number of genes, however, this disparity disappears and is even reversed. For example, on synthetically generated data with 500 cells and 500 genes, SC3 ran in 2 minutes and 9 seconds (with $k = 5$), while our algorithm ran in less than a second. The bulk of our computation comes from computing the monotonicity values, which can be independently computed, allowing for the efficient usage of parallel processing. As a result, our wall runtimes were actually 4 to 5 times shorter than the CPU times listed in Table 1.

4 Methods

4.1 Pre-processing

We filter out rare and ubiquitous genes (genes not expressed in >90% or <10% of the cells) with the methods `sc3_prepare` and `get_processed_dataset` from the SC3 R package, since such genes are likely to have little informative value [2]. The filtered matrix is then log-transformed and normalized using the `logNormCounts` method from the scater R package [13, 22]. Only the genes whose expression values were recorded across all cell subpopulations (albeit as dropout values) were analyzed because genes whose expression values were recorded in only some cells are likely to differentiate the cell subpopulations too conveniently. This condition was not required for the Yan dataset, in which all of the genes whose expression values were recorded across all cells were ubiquitous genes (presumably because expression values of 0 were preemptively removed). The data presented in the Patel dataset were already log-transformed but without adding 1 to the TPM value, so each data value x was replaced with $\log_2(2^x + 1)$.

4.2 Our Algorithm

We define our distance metric as

$$monotonicity(x, y) = \frac{\sum_{1 \le i < j \le n} \max(0, (x_j - x_i)(y_j - y_i))}{\sum_{1 \le i < j \le n} \max(0, (x_j - x_i)(y_i - y_j))}$$

where x and y represent the expression values of 2 genes over n cells. Our monotonicity value relates the positive monotonic behavior to the negative monotonic behavior. A pair of points contributes to the numerator if it exhibits positive monotonic behavior, whereas it contributes to the denominator if it exhibits negative monotonic behavior. In either case, its contribution is the area of the rectangle whose opposite corners are those 2 points.

For a pair of genes, naively computing our metric would require $O(n^2)$ time, so here we outline how we reduced the runtime to $O(n \log n)$, matching the Spearman correlation time complexity, albeit with a higher constant factor. Assume the points are

sorted by their x values. Focusing on the numerator for now, for every point j, we wish to compute

$$\sum_{\substack{i<j \\ y_i<y_j}} (x_j - x_i)(y_j - y_i)$$

Distributing the product (and using the same summation condition), we get

$$x_j y_j \left(\sum 1\right) - x_j \left(\sum y_i\right) - y_j \left(\sum x_i\right) + \left(\sum x_i y_i\right)$$

Similarly, in the denominator, for every point j, we wish to compute

$$\sum_{\substack{i<j \\ y_i>y_j}} (x_j - x_i)(y_i - y_j) = x_j \left(\sum y_i\right) - x_j y_j \left(\sum 1\right) - \left(\sum x_i y_i\right) + y_j \left(\sum x_i\right)$$

The Binary Indexed Tree (BIT) is a linear-memory data structure that offers two functions, each executed in logarithmic time: (1) to change every value in a specified prefix of an array by a specified amount and (2) to query the sum of the values in a specified prefix of the array [17]. We update and query a BIT for each of the 8 summations as we sweep through the n points from left to right, for a runtime of $O(n \log n)$ time.

Algorithm 1 Compute the monotonicity value between 2 genes

```
function MONOTONICITY(x, y)        ▷ x and y are the expression values of 2 genes over n cells
    orderX ← [1, 2, ..., n] sorted by x[i]
    orderY ← [1, 2, ..., n] sorted by y[i]
    rankY ← array of size n
    for i ← 1 to n do
        rankY[orderY[i]] ← i
    initialize onePos, xPos, yPos, xyPos, oneNeg, xNeg, yNeg, xyNeg each as a BIT of size n (with values initialized to 0)
    pos ← 0        ▷ positive monotonic behavior
    neg ← 0        ▷ negative monotonic behavior
    for i in orderX do
        k ← rankY[i]
        update onePos, xPos, yPos, xyPos with 1, x[i], y[i], x[i] · y[i], respectively, each up to index k
        update oneNeg, xNeg, yNeg, xyNeg with 1, x[i], y[i], x[i] · y[i], respectively, each up to index n + 1 − k
        pos ← pos + x[i] · y[i] · onePos.query(k) − x[i] · yPos.query(k) − y[i] · xPos.query(k) + xyPos.query(k)
        neg ← neg + x[i] · yNeg.query(k) − x[i] · y[i] · oneNeg.query(k) − xyNeg.query(k) + y[i] · xNeg.query(k)
    return pos/neg
```

Two genes with a high monotonicity value are considered to be co-expressed. It has been shown that a negative association between 2 genes' expression values can generate meaningful results regarding gene regulation [12]. However, as this paper limits itself to the application of this metric to clustering genes for subsequent cell clustering, we choose only to cluster together genes that are positively monotonic with each other. From here on, "monotonically expressed" will imply "positively monotonically expressed".

Due to the monotonic nature of our metric and the fact that the monotonicity values are not bounded, the triangle inequality is not necessarily satisfied. In other words, even if genes x and y are monotonically expressed and genes y and z are monotonically expressed, genes x and z might not be monotonically expressed. Therefore, using a conventional clustering algorithm such as k-means or hierarchical clustering with our monotonicity values would generate noisy clusters. Instead, we apply a stricter clustering algorithm which ensures that within a cluster, a randomly selected pair of genes has greater than $tightness$ probability of being monotonically expressed, for some specified probability $tightness$ (default 0.95). A pair of genes is considered monotonically expressed if its monotonicity value is greater than a specified value $threshold$ (default 15).

Algorithm 2 Cluster genes using monotonicity values

function AVGCOR($gene, cluster$)
  $res \leftarrow 0$
  for i in $cluster$ **do**
    $res \leftarrow res + cor[gene][i]$
  return $res/$SIZE($cluster$)

function CLUSTERGENES($expr$) ▷ $expr$ is an expression matrix of m genes

$$cor \leftarrow \underbrace{\begin{array}{|c|c|c|}\hline 0 & \dots & 0 \\ \hline \vdots & \ddots & \vdots \\ \hline 0 & \dots & 0 \\ \hline \end{array}}_{m} \Bigg\} m$$

  for $i \leftarrow 1$ to m **do**
    for $j \leftarrow 1$ to m **do**
      if MONOTONICITY($expr[i], expr[j]$) $> threshold$ **then**
        $cor[i][j] \leftarrow 1$
  $order \leftarrow [1, 2, \dots, m]$ sorted by decreasing STANDARDDEVIATION($expr[i]$)
  $clusters \leftarrow [\,]$
  for g in $order$ **do**
    $near \leftarrow$ the index i such that AVGCOR($g, clusters[i]$) is minimum
    if AVGCOR($g, clusters[near]$) $> tightness$ **then**
      add gene g to cluster $clusters[near]$ and update average correlations from cluster $near$ to unclustered genes
    else
      add new cluster with gene g to $clusters$ and initialize average correlations from new cluster to unclustered genes
  return $clusters$

The time complexity of computing the monotonicity value for every pair of genes is $O(m^2 n \log n)$, occupying the bulk of the computational cost. As the monotonicity values can be computed independently, parallel processing can be used to more efficiently compute them. Once the monotonicity values have been computed, the clustering algorithm can be performed in $O(m^2 \log m)$ time, potentially multiple times with various parameters. The overall time complexity is $O(m^2(n \log n + \log m))$.

4.3 Comparison of Gene Clustering Algorithms

To test the various clustering algorithms, each clustering algorithm takes in the pre-processed expression matrix and feeds a compressed feature matrix into SC3 for cell clustering. We chose SC3 because its stability makes it well-suited for testing compared to other popular cell clustering algorithms [2]. The Adjusted Rand Index (ARI) was used to evaluate how closely the resulting cell clustering matched the one provided by the original authors of the datasets [19]. To calculate the ARI between 2 clusterings of n objects, an $r \times s$ contingency table N is constructed (where the first and second clusterings generated r and s clusters, respectively) in which N_{ij} is the number of objects clustered into cluster i by the first clustering and into cluster j by the second clustering. a_i denotes $\sum_j N_{ij}$, and b_j denotes $\sum_i N_{ij}$.

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

For each dataset, SC3 was run on the unclustered expression matrix as well as the compressed feature matrix produced by each of the 6 gene clustering algorithms we tested. Our algorithm was run with its default parameters (a monotonicity threshold of 15 and a tightness value of 0.95). Clust was run with its default parameters (a tightness value of 1) and with the `-n 0` flag to skip normalization because normalization was performed in the pre-processing. K-means and hierarchical clustering require the number of clusters to be selected by the user. Several methods have been developed for determining the optimal number

of clusters for such clustering algorithms, including the elbow method, the average silhouette method, and the gap statistic method [20]. However, such methods often involve generating the clusterings for every number of clusters within a given range, which is unfeasible with the high dimensionality of scRNA-seq data. Therefore, to fairly compare our algorithm with k-means and hierarchical clustering, the latter 2 algorithms are specified to generate the same number of clusters generated by our algorithm. The scipy Python package was used to run agglomerative hierarchical clustering (with Euclidean distance and Pearson correlation distance) [18], and the scikit-learn Python package was used to run k-means clustering [5]. The Seurat R package was used to perform principal component analysis (PCA) to extract high-variance eigenvectors as cell features [23]. As pre-processing was performed beforehand, the `ScaleData` function was run with `do.scale = FALSE`. For most of the datasets we tested with, the number of cells was much smaller than the number of genes, so we used one less than the number of cells as the number of principal components. In datasets with large numbers of cells (Bozec and Shen), we used 500 principal components. For PCA, the matrix of cell embeddings was used as the compressed feature matrix. For the other gene clustering algorithms, the vectors of gene expression within each gene cluster were averaged to produce the compressed feature matrix. For any parameters not mentioned above, default values were used. An Inspiron 15 7000 laptop with an 8th generation i7 processor was used for benchmarking times.

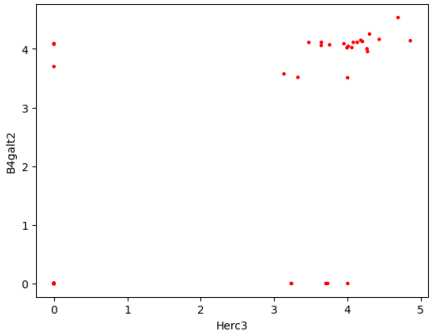
4.4 Motivation

Our expression for monotonicity is motivated by the desire for a gradual transition between positive monotonic behavior and negative monotonic behavior with regard to contribution to the numerator or denominator. In other words, suppose that Euclidean distance or Manhattan distance were used instead. If $x_i \simeq x_j$ and $y_j >> y_i$, a slight shift from $x_i < x_j$ to $x_i > x_j$ would move a large contribution from the numerator to the denominator, or vice versa. On the other hand, our metric gives only minor weight to 2 points who deviate significantly in one dimension but insignificantly in the other dimension, so a shift from $x_i < x_j$ to $x_i > x_j$ would allow the numerator contribution to gradually reach 0 before allowing the denominator contribution to gradually increase from 0. Meanwhile, if the points differ significantly in both dimensions, we are more confident in their positive or negative monotonic behavior, so we grant them more contribution to the numerator or denominator, respectively. With randomly generated data, the expected monotonicity value is 1. It is worth noting that although the magnitudes of the values are factored into our metric, arbitrarily adjusting the scale of the values would impact the numerator and denominator by the same factor, leading our distance metric to be scale-invariant, as are many commonly used metrics.

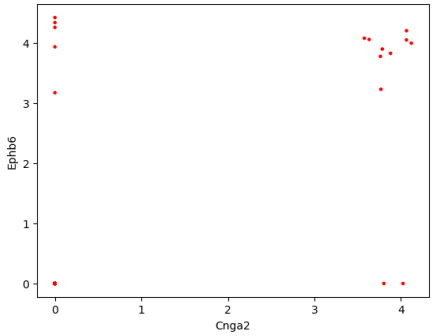
Our metric's ability to take into account the magnitude of values comes at the cost of allowing outliers to potentially sway the monotonicity. As a result, pre-processing methods are necessary to transform the distribution of the data for more balanced monotonicity values. When clustering genes based on a monotonicity-based metric, it is important to note that uniformly expressed genes (with either very scarce dropout values or very abundant dropout values) are dangerous because they are likely to have high monotonicity values with many genes, even when a meaningful relationship may not exist. As a result, the pre-processing performed by the aforementioned SC3 methods is of critical importance here in filtering out such genes. Additionally, in the clustering process, we iterate through the genes in order of decreasing variance for two reasons: (1) Even among the genes that pass the pre-processing stage, it can be expected for there to be some relatively uniformly expressed genes. When these genes are involved, the monotonicity value would be volatile because both the numerator and denominator would be small. Considering uniformly expressed genes (with low variance) at the end would prevent them from swaying the average correlation values; and since the ultimate clustered expression values are obtained by averaging the gene vectors of expression values within each cluster, uniformly expressed genes will have little impact on the clustered expression values because they will have relatively uniform influence over the values of all cells. (2) Due to the high dimensionality of scRNA-seq data, genes with high variances, which will tend to serve as the cluster "centroids", will tend to be well-separated.

4.5 Sample Data

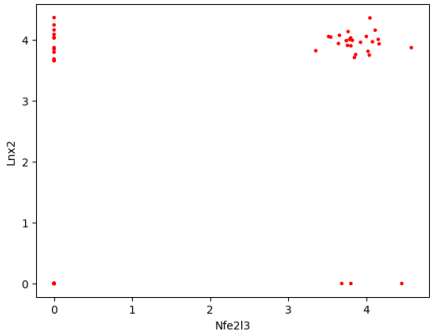
The graphs below plot pairs of pre-processed gene expression data from the Biase dataset [24], featuring 49 cells, represented as points in the graphs. These sample data help illustrate how our monotonicity metric more effectively quantifies the degree of co-expression between genes compared to commonly used metrics. Here, we focus on the Pearson, Spearman, and Kendall tau-b correlations because they are scale-invariant and deterministic, allowing their performance to be independent of the number of data points and how the data are scaled [14].



Although several dropout values don't coincide, the vast majority of the data is monotonic, resulting in a monotonicity value of 22.4. However, the co-expression is not so evident in the Pearson coefficient (0.678), Spearman coefficient (0.734), or tau-b coefficient (0.607).



Most of the points are piled up near (0, 0), so only 7 out of the 49 pairs of expression values don't coincide, resulting in a monotonicity value of 20.8. For comparison, the Pearson coefficient is 0.617, the Spearman coefficient is 0.574, and the tau-b coefficient is 0.511.



A significant number of dropout values don't coincide, resulting in a monotonicity value of only 5.1. For comparison, the Pearson coefficient is 0.379, the Spearman coefficient is 0.259, and the tau-b coefficient is 0.185.

5 Code Availability

Our software is available at https://github.com/dubai03nsr/Monotonic-Gene-Clustering.

6 Data Availability

Biase — The 3 cell labels used were zygote, 2-cell, and 4-cell.
Shen — The 3 cell labels used were wild type (WT), Jα18-deficient, and CD1d-deficient mice.
Yan — The 8 cell labels used were oocyte, zygote, 2-cell embryo, 4-cell embryo, 8-cell embryo, morulae, late blastocyst, and human embryonic stem cell (hESC).
Goolam — The 5 cell labels used were 2-cell, 4-cell, 8-cell, 16-cell, and 32-cell.
Deng — The 10 cell labels used were zygote, early 2-cell, mid 2-cell, late 2-cell, 4-cell, 8-cell, 16-cell, earlyblast, midblast, and lateblast.
Pollen — The 11 cell labels used were 2338, 2339, K562, BJ, HL60, iPS, Kera, GW21.2, GW21, NPC, and GW16.
Tatsuoka — The 2 cell labels used were young and old.
Treutlein — The 4 cell labels used were E14.5, E16.5, E18.5, and adult.
Patel — The 5 cell labels used were MGH26, MGH28, MGH29, MGH30, and MGH31.
Bozec — The 3 cell labels used were Lung_OVA+SIA, Synovium_OVA+SIA, and Synovium_SIA.

7 Author Contributions

V.W. conceived and implemented the algorithms, performed testing, and wrote the manuscript. P.A.C. guided the project and revised the manuscript. C.M. oversaw the project.

8 References

[1] Wolf, F., Angerer, P. & Theis, F. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol 19, 15 (2018). https://doi.org/10.1186/s13059-017-1382-0

[2] Kiselev, V., Kirschner, K., Schaub, M. et al. SC3: consensus clustering of single-cell RNA-seq data. Nat Methods 14, 483–486 (2017). https://doi.org/10.1038/nmeth.4236

[3] Pang, Minxing & Tegner, Jesper. (2020). Multitask learning for Transformers with application to large-scale single-cell transcriptomes. https://doi.org/10.1101/2020.02.05.935239.

[4] Bellman, Richard Ernest; Rand Corporation (1957). Dynamic programming. Princeton University Press. p. ix. ISBN 978-0-691-07951-6.

[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[6] Kiselev, V.Y., Andrews, T.S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. Nat Rev Genet 20, 273–282 (2019). https://doi.org/10.1038/s41576-018-0088-9

[7] Gangman Yi, Sing-Hoi Sze, Michael R. Thon, Identifying clusters of functionally related genes in genomes, Bioinformatics, Volume 23, Issue 9, 1 May 2007, Pages 1053–1060, https://doi.org/10.1093/bioinformatics/btl673

[8] Abu-Jamous, B., Kelly, S. Clust: automatic extraction of optimal co-expressed gene clusters from gene expression data. Genome Biol 19, 172 (2018). https://doi.org/10.1186/s13059-018-1536-8

[9] Sipko van Dam, Urmo Võsa, Adriaan van der Graaf, Lude Franke, João Pedro de Magalhães, Gene co-expression analysis for functional classification and gene–disease predictions, Briefings in Bioinformatics, Volume 19, Issue 4, July 2018, Pages 575–592, https://doi.org/10.1093/bib/bbw139

[10] Brennecke, P., Anders, S., Kim, J. et al. Accounting for technical noise in single-cell RNA-seq experiments. Nat Methods 10, 1093–1095 (2013). https://doi.org/10.1038/nmeth.2645

[11] Zwillinger, D. and Kokoska, S. (2000). CRC Standard Probability and Statistics Tables and Formulae. Chapman & Hall: New York. 2000. Section 14.7

[12] Xiangjun Zhou, Manrong Zha, Jing Huang, Li Li, Muhammad Imran, Cankui Zhang, StMYB44 negatively regulates phosphate transport by suppressing expression of PHOSPHATE1 in potato, Journal of Experimental Botany, Volume 68, Issue 5, 1 February 2017, Pages 1265–1281, https://doi.org/10.1093/jxb/erx026

[13] Amezquita R, Lun A, Becht E, Carey V, Carpp L, Geistlinger L, Marini F, Rue-Albrecht K, Risso D, Soneson C, Waldron L, Pages H, Smith M, Huber W, Morgan M, Gottardo R, Hicks S (2020). "Orchestrating single-cell analysis with Bioconductor." _Nature Methods_, *17*, 137-145. <URL: https://www.nature.com/articles/s41592-019-0654-x>.

[14] Hauke, J., & Kossowski, T. (2011). Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data, Quaestiones Geographicae, 30(2), 87-93. doi: https://doi.org/10.2478/v10117-011-0021-1

[15] Botía, J.A., Vandrovcova, J., Forabosco, P. et al. An additional k-means clustering step improves the biological features of WGCNA gene co-expression networks. BMC Syst Biol 11, 47 (2017). https://doi.org/10.1186/s12918-017-0420-6

[16] Pagnuco IA, Pastore JI, Abras G, Brun M, Ballarin VL. Analysis of genetic association using hierarchical clustering and cluster validation indices. Genomics. 2017 Oct;109(5-6):438-445. doi: 10.1016/j.ygeno.2017.06.009. Epub 2017 Jul 8. PMID: 28694080.

[17] Fenwick, P.. "A new data structure for cumulative frequency tables." Software: Practice and Experience 24 (1994): n. pag.

[18] Virtanen, P., Gommers, R., Oliphant, T.E. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2

[19] Steinley, D., Brusco, M. J., & Hubert, L. (2016). The variance of the adjusted Rand index. Psychological Methods, 21(2), 261–272. https://doi.org/10.1037/met0000049

[20] Tibshirani, R., Walther, G. and Hastie, T. (2001). Estimating the number of data clusters via the Gap statistic. Journal of the Royal Statistical Society B, 63, 411–423.

[21] Rozenblatt-Rosen O, Stubbington MJT, Regev A, Teichmann SA. The Human Cell Atlas: from vision to reality. Nature. 2017 Oct 18;550(7677):451-453. doi: 10.1038/550451a. PMID: 29072289.

[22] McCarthy DJ, Campbell KR, Lun ATL, Willis QF (2017). "Scater: pre-processing, quality control, normalisation and visualisation of single-cell RNA-seq data in R." _Bioinformatics_, *33*, 1179-1186. doi: 10.1093/bioinformatics/btw777

[23] Butler, A., Hoffman, P., Smibert, P. et al. Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat Biotechnol 36, 411–420 (2018). https://doi.org/10.1038/nbt.4096

[24] Biase, Fernando H et al. "Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing." Genome research vol. 24,11 (2014): 1787-96. doi:10.1101/gr.177725.114

[25] Shen H, Gu C, Liang T, Liu H, Guo F, Liu X. Unveiling the heterogeneity of NKT cells in the liver through single cell RNA sequencing. Sci Rep. 2020 Nov 10;10(1):19453. doi: 10.1038/s41598-020-76659-1. PMID: 33173202; PMCID: PMC7655820.

[26] Yan L, Yang M, Guo H, Yang L, Wu J, Li R, Liu P, Lian Y, Zheng X, Yan J, Huang J, Li M, Wu X, Wen L, Lao K, Li R, Qiao J, Tang F. Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. Nat Struct Mol Biol. 2013 Sep;20(9):1131-9. doi: 10.1038/nsmb.2660. Epub 2013 Aug 11. PMID: 23934149.

[27] Goolam M, Scialdone A, Graham SJL, Macaulay IC, Jedrusik A, Hupalowska A, Voet T, Marioni JC, Zernicka-Goetz M. Heterogeneity in Oct4 and Sox2 Targets Biases Cell Fate in 4-Cell Mouse Embryos. Cell. 2016 Mar 24;165(1):61-74. doi: 10.1016/j.cell.2016.01.047. PMID: 27015307; PMCID: PMC4819611.

[28] Deng Q, Ramsköld D, Reinius B, Sandberg R. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. Science. 2014 Jan 10;343(6167):193-6. doi: 10.1126/science.1245316. PMID: 24408435.

[29] Pollen, A., Nowakowski, T., Shuga, J. et al. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. Nat Biotechnol 32, 1053–1058 (2014). https://doi.org/10.1038/nbt.2967

[30] Tatsuoka, Hisato et al. "Single-Cell Transcriptome Analysis Dissects the Replicating Process of Pancreatic Beta Cells in Partial Pancreatectomy Model." iScience vol. 23,12 101774. 6 Nov. 2020, doi:10.1016/j.isci.2020.101774

[31] Andreev D, Liu M, Kachler K, Llerins Perez M, Kirchner P, Kölle J, Gießl A, Rauber S, Song R, Aust O, Grüneboom A, Kleyer A, Cañete JD, Ekici A, Ramming A, Finotto S, Schett G, Bozec A. Regulatory eosinophils induce the resolution of experimental arthritis and appear in remission state of human rheumatoid arthritis. Ann Rheum Dis. 2020 Nov 4:annrheumdis-2020-218902. doi: 10.1136/annrheumdis-2020-218902. Epub ahead of print. PMID: 33148700.

[32] Treutlein, B., Brownfield, D., Wu, A. et al. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. Nature 509, 371–375 (2014). https://doi.org/10.1038/nature13173

[33] Patel AP, Tirosh I, Trombetta JJ, Shalek AK, Gillespie SM, Wakimoto H, Cahill DP, Nahed BV, Curry WT, Martuza RL, Louis DN, Rozenblatt-Rosen O, Suvà ML, Regev A, Bernstein BE. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. Science. 2014 Jun 20;344(6190):1396-401. doi: 10.1126/science.1254257. Epub 2014 Jun 12. PMID: 24925914; PMCID: PMC4123637.