ClusterDE: a post-clustering differential expression method

Carson Zhang

Contributing authors: carson.zhang@campus.lmu.de;

Abstract

In typical differential expression analysis, a clustering algorithm is applied to scRNA-seq data, and then a differential expression test is conducted in order to identify genes that are differentially expressed between the clusters. However, this procedure constitutes "double dipping", as it first clusters the data to identify cell types, and then uses those same clusters to identify cell-type marker genes. This leads to an inflated FDR for DE genes. Song et al. propose ClusterDE (Song et al. 2023), a post-clustering DE method that controls the FDR of DE genes. ClusterDE generates a synthetic null dataset that preserves the structure of the real data, computes differences between this null dataset and the real data, then performs FDR control on the results. Simulations and real data analysis demonstrate that ClusterDE controls the FDR and identifies cell-type marker genes as top DE genes, successfully distinguishing them from housekeeping genes. Furthermore, investigation of the covariance

Table of contents

Introduction

Cell-type annotation

Motivation

Understanding which types of cells are in a data sample allows an analyst to better make use of existing knowledge about those cells. “Cell annotation” is the process of labeling cells in a sample of data. In this paper, the focus is on annotating the “cell type” of each cell: a cellular phenotype that is robust across datasets (Heumos et al. 2023). For example, plasma B cells are one type of white blood cell that are involved in the human body’s immune response by secreting antibodies (Heumos et al. 2023). T cells are another type of white blood cell that are also involved in immune response. They produce “cytokines, which are signaling proteins that activate other parts of the human immune system” (Green et al. 2024). A scientist interested in a patient’s immune response may be interested in the counts of B cells and T cells (and their subtypes): for example, in order to better understand the roles of each cell, or how they affect patient outcomes. Cell type annotation is required in order to obtain this information from e.g. a blood sample.

Distinction between different types of T cells requires better annotation methods.

Discussion of terminology (type vs. identity).

Manual annotation using cell type markers

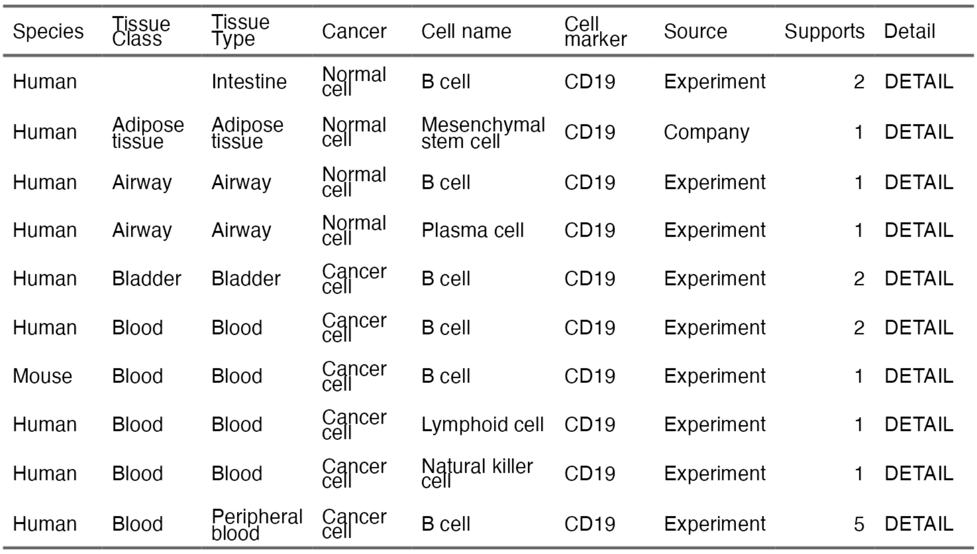
To perform this annotation, it is common to use marker genes. In an idealized setting, a cell type would have a unique marker: a single gene such that when this gene is highly expressed in a cell, we are confident that the cell has a given type. However, this is often unrealistic or unachievable: therefore, we seek some some combination of marker genes that, when taken together, identify a cell type.

Consider the gene CD19. It is a marker for 220 human cell types, and 66 mouse cell types Figure 15. Furthermore, Figure 1 demonstrates how a single gene may be a marker for related cell types. Suppose that in a sample of human blood, one observes both cancerous B cells and cancerous Lymphoid cells. The expression level of CD19 alone would not help an analyst differentiate between the cell types. Therefore, it is important to discover multiple marker genes, especially for classifying similar or related cell types.

These marker genes are compiled into databases such as CellMarker 2.0, a database containing, at the time of initial publication, 26,915 cell marker genes for 2578 cell types (Hu et al. 2022). Such databases are essential to manual annotation using cell markers.

Automated annotation

It is also possible to annotate cells in an automated manner. These automatic methods might also use marker genes, similar to manual annotation. They may also be



Species	Tissue Class	Tissue Type	Cancer	Cell name	Cell marker	Source	Supports	Detail
Human		Intestine	Normal cell	B cell	CD19	Experiment	2	DETAIL
Human	Adipose tissue	Adipose tissue	Normal cell	Mesenchymal stem cell	CD19	Company	1	DETAIL
Human	Airway	Airway	Normal cell	B cell	CD19	Experiment	1	DETAIL
Human	Airway	Airway	Normal cell	Plasma cell	CD19	Experiment	1	DETAIL
Human	Bladder	Bladder	Cancer cell	B cell	CD19	Experiment	2	DETAIL
Human	Blood	Blood	Cancer cell	B cell	CD19	Experiment	2	DETAIL
Mouse	Blood	Blood	Cancer cell	B cell	CD19	Experiment	1	DETAIL
Human	Blood	Blood	Cancer cell	Lymphoid cell	CD19	Experiment	1	DETAIL
Human	Blood	Blood	Cancer cell	Natural killer cell	CD19	Experiment	1	DETAIL
Human	Blood	Peripheral blood	Cancer cell	B cell	CD19	Experiment	5	DETAIL

Figure 1: Some cells marked by the CD19 gene.

traditional supervised machine learning classifiers. However, the quality of automated annotations depends on many factors, such as the quality of the training data and its similarity to the data that you ultimately want to classify. Heumos et al. describe and discuss automated annotation methods in much more detail (Heumos et al. 2023).

Differential expression testing

Differential expression testing is the primary method by which scientists identify marker genes. If gene A is differentially expressed across two conditions, then it may be a good candidate for a marker gene.

The double-dipping issue

However, when the two conditions are two clusters, we have an issue: we used our data twice. First, we clustered into two groups. Then, we tested for variation in gene expression levels between those groups. This is problematic because we found variation in the data that may be spurious, and then we tested for variation *that we already know is there*, leading to invalid inference.

The Seurat default DE test is susceptible to this double-dipping, and comes with a warning in one of the vignettes: "the p-values obtained from this analysis should be interpreted with caution" (Hao et al. 2021). This illustrates that this is a known issue that is still easy to fall prey to if an analyst is not careful.

Toy example illustrating double dipping

The example comes from the ClusterDE paper (see Figure S15, (Song et al. 2023)). Suppose we only have 2 genes. However, they come from a single homogeneous population (in this case, the data is generated from a single multivariate normal distribution).

Now, pretend that we don't know that the data is homogeneous. We decide to cluster the data into 2 groups, since we want to perform differential expression testing. We observe the clustering in Figure 2. Because we forced the clustering algorithm to find two groups (in this case, ran k-means with $k = 2$), the algorithm found some variation (along the max-variance direction) where none truly exists.

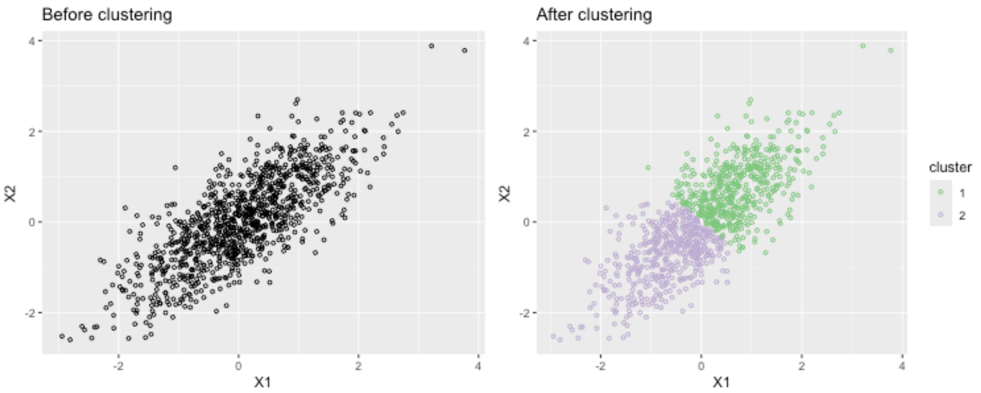
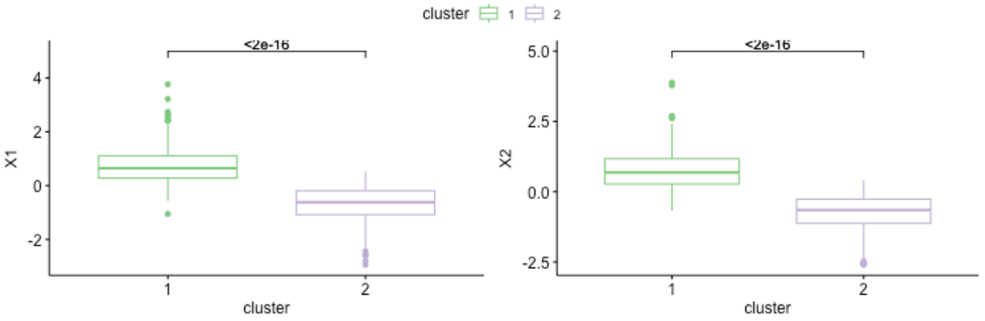


Figure 2

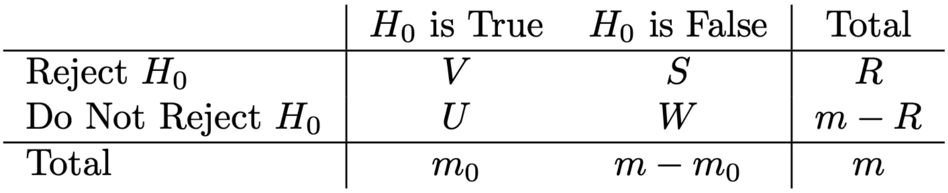
Next, we perform differential expression testing for each gene, and compute very low p-values. Therefore, we declare that both genes $X1$ and $X2$ are differentially expressed, when in reality they come from the same cell population. Why do we get such low p-values when the null hypothesis is true? Since our inference is conditional on the clusters, and the clustering algorithm found spurious variation, the clustering encourages discovery of DE genes even when none exist. The same principle generalizes when we have thousands of genes, as is the case in realistic datasets.



False discoveries

As we are conducting m hypothesis tests, we are in a multiple testing situation. Consider the notation defined in Figure 3 for the types and numbers of possible decisions.

Let $\frac{V}{R}$ be the **false discovery proportion**. While this already looks like the thing we want to control, it is impossible to control directly. This is because when we are given a single particular dataset, we have no guarantees about the values of V and F, i.e. which hypotheses are true or false. If we had such guarantees, then we wouldn't need to perform hypothesis tests in the first place. However, we can control the expected false discovery proportion, which we call the **false discovery rate** $E[\frac{V}{R}]$ (James et al. 2021).



	H_0 is True	H_0 is False	Total
Reject H_0	V	S	R
Do Not Reject H_0	U	W	$m - R$
Total	m_0	$m - m_0$	m

Figure 3: A table defining notation for various decisions resulting from hypothesis tests. Table 13.2 from ISLR (James et al. 2021).

Definition (false discovery rate):

$$\text{FDR} := E[\frac{\text{number of false rejections}}{\text{total number of rejections} \vee 1}]$$

In the above definition, the $\vee$ operator takes the maximum of the left and right expressions. This avoids dividing by 0 when we do not reject any null hypotheses. For simplicity, we may omit this operator in the rest of this paper.

Notation

We observe a cell × gene count matrix with n rows (cells) and m columns (genes).

Definition (count matrix): the **count matrix** $\mathbf{X} \in \mathbb{N}_{0+}^{n \times m}$ is defined as

$$\mathbf{X} := \begin{bmatrix} X_{11} & \dots & X_{1m} \\ \vdots & \ddots & \\ X_{n1} & & X_{nm} \end{bmatrix}$$

.

The goal is to find $Z \in \{0, 1\}^n$ (recall that ClusterDE can only help one differentiate between two cell types). In an ideal world, we would already know Z.

The **idealized count matrix $\mathbf{X}|\hat{Z}$** is defined as

$$\mathbf{X}|Z := \begin{bmatrix} X_{11} & \dots & X_{1m} & Z_1 \\ \vdots & \ddots & & \vdots \\ X_{n1} & & X_{nm} & Z_n \end{bmatrix}$$

.

However, we can only approximate Z through clustering, since we do not know the cell types in advance (otherwise, we would not have to do any annotation!).

The **clustered count matrix $\mathbf{Y}|\hat{Z}$** is defined as

$$\mathbf{X}|\hat{Z} := \begin{bmatrix} X_{11} & \dots & X_{1m} & \hat{Z}_1 \\ \vdots & \ddots & & \vdots \\ X_{n1} & & X_{nm} & \hat{Z}_n \end{bmatrix}$$

Double-dipping

We want to test the following idealized null hypothesis.

$$H_{0j} : \mu_{Z=0,j} = \mu_{Z=1,j}$$

However, we can only test the double-dipping null hypothesis, since in the clustered count matrix, we do not observe Z.

$$H_{0j}^{DD} : \mu_{\hat{Z}=0,j} = \mu_{\hat{Z}=1,j}$$

False discoveries occur when the idealized null hypothesis does not hold, but the double-dipping null hypothesis holds. In other words, false discoveries occur when we made the right decision for our hypothesis test, but we set up the wrong test. Thus, in a naive differential expression test, we are overly reliant on $\hat{Z}$ being a good approximation of Z.

Song, et al. propose ClusterDE as a way to control the false discovery rate in differential expression testing.

ClusterDE

Summary of steps

The ClusterDE method consists of four basic steps, summarized in Figure 4.

1. Generate a synthetic null dataset that consists of a single cluster but otherwise mimics the real data.
2. Separately for each dataset, cluster the cells into two groups.
3. Separately for each dataset, perform differential expression testing between the two groups from step 2.

4. Combine the results to determine which genes to output as discoveries (DE genes).

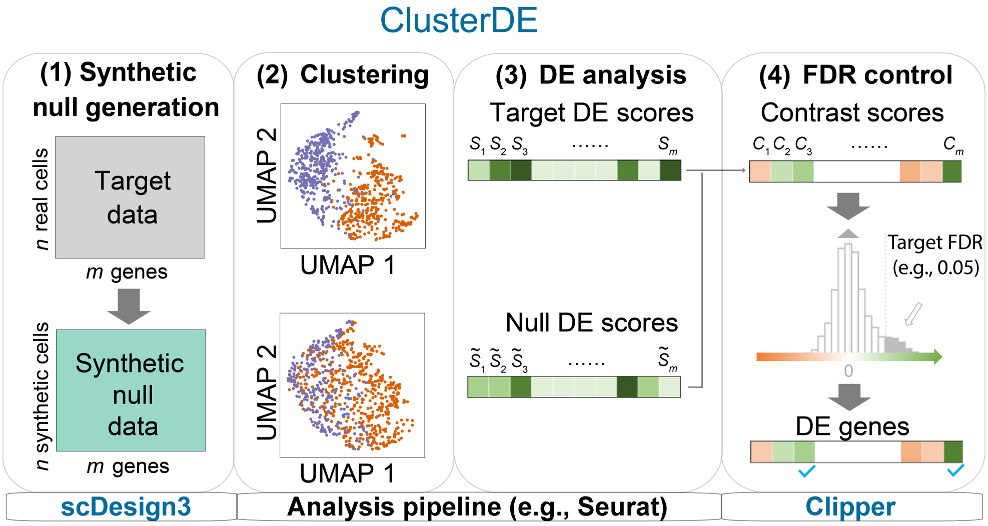


Figure 4: A visual overview of the ClusterDE method. In step 1, a negative control dataset is generated. In step 2, a clustering algorithm is applied to each dataset. In step 3, a differential expression test is performed for each gene, computing a DE score for each gene in each dataset. In step 4, the difference in results is computed as a contrast score, and Clipper is used to choose a minimum contrast score for the true DE genes outputted by ClusterDE.

Step 1: synthetic null generation

Idea: negative control

The idea of using a synthetic dataset to represent the null hypothesis comes from the broader idea of negative control. To illustrate this

Generating a negative control dataset using a copula

To actually generate this negative control data, (Song et al. 2023) use the copula approach. This is because statistical packages such as `R` do not come with samplers already implemented, so special methods are required to simulate data from the desired multivariate negative binomial distribution. Thus, ClusterDE uses the copula-based sampler implemented in scDesign3 (Song et al. 2024) for its *in silico* negative control data: that is, data that was created by a computer (Ekins, Mestres, and Testa 2007). We describe some of the mathematics underlying copulas.

Theorem (Probability Integral Transform): $Y := F_X(X) \sim \text{Uniform}(0, 1)$.

Proof:

$$
\begin{aligned}
F_Y(y) &= P(Y \leq y) \\
&= P(F_X(X) \leq y) && \text{(substituted the definition of } Y\text{)} \\
&= P(X \leq F_X^{-1}(y)) && \text{(applied } F_X^{-1} \text{ to both sides)} \\
&= F_X(F_X^{-1}(y)) && \text{(the definition of a CDF)} \\
&= y
\end{aligned}
$$

Proof from the Wikipedia page ("Probability Integral Transform," n.d.), annotations from a blog post (Zhang 2023). A more rigorous proof and discussion can be found in Theorem 2.1.10 in Casella and Berger (Casella and Berger 2001).

Intuition for the PIT

One can imagine drawing the distribution of $F_X(X)$ one section at a time, dealing with intervals that are between known values of $F_X(X)$ (i.e. between quantiles). Since values of X between the p and q-quantiles (with $p < q$) occur with probability $q - p$, and the distance along the horizontal axis of the density plot of $F_X(X)$ is $q - p$, it follows that the value of the density of function of $F_X(X)$ is always $\frac{q-p}{q-p} = 1$. A more detailed discussion of this idea can be found in the blog (Zhang 2023).

The main takeaway is that if we can compute F^{-1}, we can move freely between a standard uniform random variable and a random variable with distribution F. Sklar's Theorem, and therefore the copula approach to modeling multivariate distributions, relies on this result.

Theorem (Sklar's Theorem): Let $\mathbf{X}$ be a m-dimensional random vector with joint cumulative distribution function F and marginal distribution functions $F_j, j = 1, ..., m$. The joint CDF can be expressed as

$$F(x_1, ..., x_m) = C(F_1(x_1), ..., F_m(x_m))$$

with associated probability density (or mass) function

$$f(x_1, ..., x_m) = c(F_1(x_1), ..., F_m(x_m))f_1(x_1)...f_m(x_m)$$

for a m-dimensional copula C with copula density c.

The inverse also holds: the copula corresponding to a multivariate CDF F with marginal distribution functions $F_j, j = 1, ..., m$ can be expressed as

$$C(u_1, ..., u_m) = F(F_1^{-1}(u_1), ..., F_m^{-1}(u_m))$$

, and the copula density (or mass) function is

$$c(u_1, ..., u_m) = \frac{f(F_1^{-1}(u_1), ..., F_m^{-1}(u_m))}{f_1(F_1^{-1}(u_1))...f_m(F_m^{-1}(u_m))}$$

.

(The theorem statement and notation is from Czado (Czado 2019).)

Proof: See Nelsen (Nelsen 2006).

Sklar's Theorem allows statisticians to use the copula approach to model the joint distribution: the goal is now to find a copula C that yields a good approximation of F. ClusterDE makes the popular choice of the Gaussian copula to model the multivariate gene distribution, which is convenient because it has existing software implementations [see mvtnorm, numpy]. Then, to generate data from F using the copula C, we can perform the following steps:

- Estimate the correlation matrix of the target distribution (see Figure 6). This requires a transformation of the discrete CDF values into continuous $U(0, 1)$ variables: $U_{ij} = V_{ij}\hat{F}_j(Y_{ij}) + (1 - V_{ij})\hat{F}_j(Y_{ij})$ (Song et al. 2023).
- Sample $\tilde{Z} \in \mathbb{R}^{n \times m}$ from the multivariate Gaussian distribution with a correlation (or covariance) matrix that matches the target distribution (see Figure 5).
- Compute the Gaussian CDF to transform the marginal distributions into standard uniform marginals.
- Compute the inverse negative binomial CDF to transform the uniform marginal distributions into negative binomial distributions (see Equation 1). The correlation structure from the original multivariate Gaussian data will be preserved.

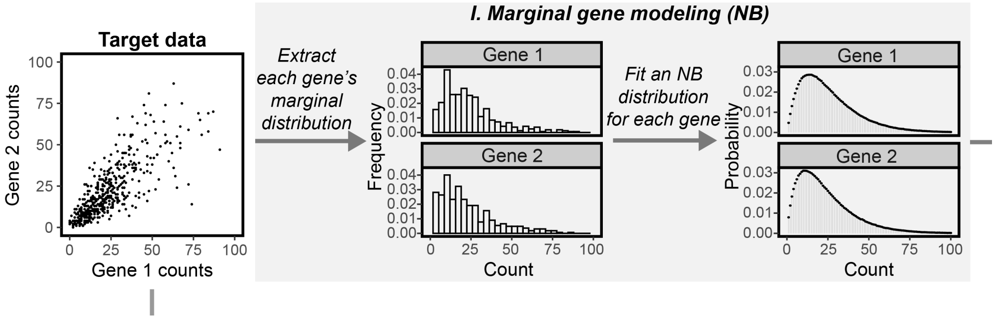


Figure 5: Fit a marginal distribution for each gene. The copula approach allows us to model the marginal distributions separately from the covariance structure of the variables (see Figure 6) (Song, Li, and Chen 2024).

$$\begin{bmatrix} \hat{F}_1^{-1}(\Phi(\tilde{Z}_{11})) & \dots & \hat{F}_m^{-1}(\Phi(\tilde{Z}_{1m})) \\ \vdots & \ddots & \\ \hat{F}_1^{-1}(\Phi(\tilde{Z}_{n1})) & & \hat{F}_m^{-1}(\Phi(\tilde{Z}_{nm})) \end{bmatrix} = \begin{bmatrix} \tilde{X}_{11} & \dots & \tilde{X}_{1m} \\ \vdots & \ddots & \\ \tilde{X}_{n1} & & \tilde{X}_{nm} \end{bmatrix} \tag{1}$$

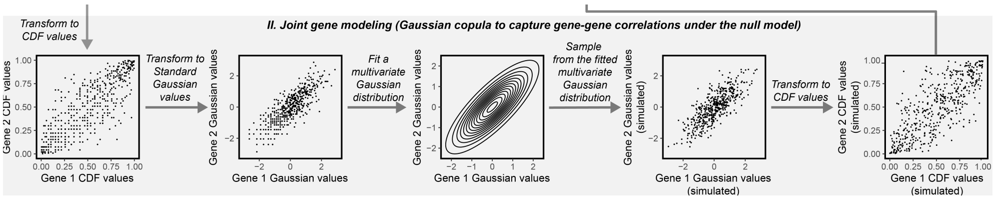

Transform to CDF values
II. Joint gene modeling (Gaussian copula to capture gene-gene correlations under the null model)
Gene 2 CDF values
Gene 1 CDF values
0.00
0.25
0.50
0.75
1.00
Transform to Standard Gaussian values
Gene 2 Gaussian values
Gene 1 Gaussian values
−2
0
2
Fit a multivariate Gaussian distribution
Gene 2 Gaussian values
Gene 1 Gaussian values
−2
−1
0
1
2
Sample from the fitted multivariate Gaussian distribution
Gene 2 Gaussian values (simulated)
Gene 1 Gaussian values (simulated)
−2
0
2
Transform to CDF values
Gene 2 CDF values (simulated)
Gene 1 CDF values (simulated)
0.00
0.25
0.50
0.75
1.00

Figure 6: Estimate the covariance matrix for the m-dimensional gene distribution. The copula approach allows us to model the correlations between genes separately from their marginal distributions (see Figure 5) (Song, Li, and Chen 2024).

Step 2: Clustering

This is the usual clustering step in differential expression testing. We perform clustering for both the target (real) data and synthetic null data.

The generally-accepted current best practice is to use the Leiden algorithm to cluster scRNA-seq data (Heumos et al. 2023). Traag et al. demonstrated that it outperforms the older Louvain algorithm in both clustering quality and computation running time (Traag, Waltman, and Eck 2019).

The Leiden algorithm is supported by both `Scanpy` and `Seurat`. However, since Louvain is still the `Seurat` default, and there is currently not a fully-featured `R` implementation, some of the analyses discussed in this paper (especially those that come from the ClusterDE paper) may use the Louvain algorithm.

Step 3: DE testing

DE tests are performed as usual on the two clusters.

Define Wilcoxon rank-sum test. `presto` package in R.

Step 4: false discovery rate control using Clipper

In Step 4, ClusterDE uses the Clipper method to choose the discoveries from step 3 to output as true DE genes.

Intuition

Given that the negative control generated in step 1 accomplished its goal, the two datasets should be similar, and therefore the p-values (and DE scores) outputted by each test should be similar. This means that, when a test on a given gene has a very low p-value, but this p-value is similar across both datasets, it is reasonable to believe that this low p-value occurred due to noise. However, when a p-value is much lower in the real data than in the synthetic null data, this indicates that the gene is truly differentially expressed between the two clusters.

Clipper

As input, Clipper takes contrast scores $C_1, ..., C_m$ (one for every gene). We define the contrast score below.

Definition (DE score): Let P_j be the p-value resulting from the target data DE test on the j-th gene, $j = 1, ..., m$. Then, the DE score on the target data is defined as

$$S_j := -\log_{10}(P_j)$$

.

Furthermore, let $\tilde{P}_j$ be the p-value resulting from the null data DE test on the j-th gene, $j = 1, ..., m$. Then, the DE score on the null data is defined as

$$\tilde{S}_j = -\log_{10}(\tilde{P}_j)$$

.

Definition (contrast score): the contrast score for gene j is defined as

$$C_j = S_j - \tilde{S}_j$$

From this definition, we can see that high contrast scores correspond to more confident discoveries. This follows from the fact that lower p-values on the target data indicate

We note the similarity of the DE score definition, and therefore interpretation, to the S-value (Rafi and Greenland 2020). The S-value of a test is defined as $-\log_2(P)$, where P is the p-value of the test. This can be interpreted as the amount of information the test contains against the null hypothesis, or alternatively, how surprising the test result is under the null hypothesis, with larger values indicating more information and more surprise. To see this, consider that the logarithm (whether base 2 or 10) of a smaller p-value will be a negative number with higher magnitude, and the negative sign turns this into a larger positive number. The DE score S_j is $\log_{10}(2)$ times the S-value, so the interpretations still hold.

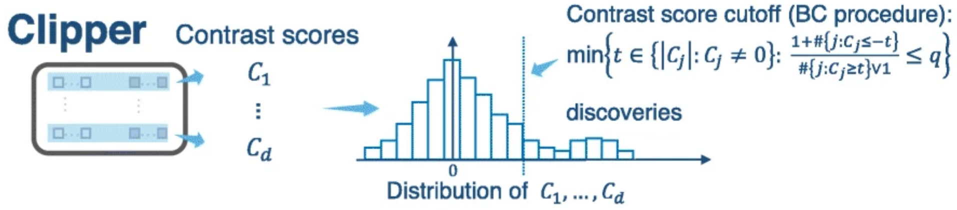

Clipper
Contrast scores
C_1
⋮
C_d
Contrast score cutoff (BC procedure):
$\min\left\{t \in \{|C_j| : C_j \neq 0\} : \frac{1+\#\{j : C_j \leq -t\}}{\#\{j : C_j \geq t\} \vee 1} \leq q\right\}$
discoveries
0
Distribution of $C_1, ..., C_d$

Figure 7: The Clipper method for FDR control. Part of Figure 1.b. from Ge et al. (Ge et al. 2021).

Definition (Clipper cutoff): Clipper chooses the minimum positive contrast score t^* to output as a discovery as follows.

$$t* = \min\left\{ t \in \{|C_j| : C_j \neq 0\} : \frac{1 + \#\{j : C_j \leq -t\}}{\#\{j : C_j \geq t\} \vee 1} \leq q \right\}$$

Intuition for the Clipper cutoff

Figure 8 illustrates the intuition behind the Clipper cutoff. For our m hypothesis tests, there are two cases:

1. The null hypothesis is true. Then the contrast score distribution is symmetric: there is no reason to assume that the target data has systematically more surprising results than the synthetic null data, since both datasets contain only a single cell type.
2. The null hypothesis is false. Then there are truly multiple cell types in the target data. However, we still have only one cell type in the synthetic null data (by construction). Therefore, we should expect to find more true differentially expressed genes, and the differences should be greater in the target data; after all, there is true variation in the target data.

Recall that we are only going to consider some section of the right tail of this distribution as discoveries. How can we choose which section?

Because this contrast score distribution is symmetric,

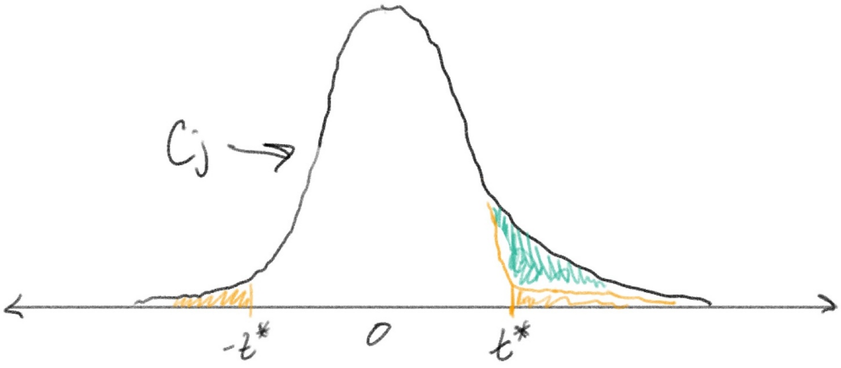


Figure 8: Motivation for the choice of contrast score cutoff. The orange tails represent the distribution under the null hypothesis. Since this distribution is symmetric, we know the rest of the right tail represents true discoveries, and we can use this symmetry to estimate the FDR at a given threshold value.

Differential expression methods that address double-dipping

Count splitting

TN Test

"Traditional" FDR control methods

(mention FDR control like Benjamini-Hochberg)

Considerations for using ClusterDE in practice

Symmetry assumption for contrast scores

In step 4, the Clipper method for FDR control assumes that the contrast score distribution is symmetric. In practice, this symmetry assumption may be violated. ClusterDE tests the symmetry of the contrast score distribution using Yuen's trimmed mean test: if the test statistic has p-value < 0.001, reject the null hypothesis of symmetry, and perform a contrast score adjustment. It uses a one-sided "greater than" hypothesis for this test: that is, it only adjusts the contrast scores when too few contrast scores are negative. This is because the authors wanted to be conservative with their adjustment strategy, only transforming the contrast scores when they know that there would have been too many false discoveries. When there are too many negative contrast scores, these will not lead to an inflated false discovery rate, since only positive contrast scores become discoveries.

The software implementation for Yuen's trimmed mean test used by the authors comes from the `PairedData` `R` package.

How to handle multiple clusters

ClusterDE is only designed to handle two cell clusters. Therefore, the authors recommend the following steps in the presence of multiple clusters (Song et al. 2023):

1. Find two clusters that look ambiguous. If you have prior knowledge, feel free to use it to manually choose these two clusters: a UMAP plot can aid in this process (see Figure 9). If you want to do this computationally, this can be accomplished by running `Seurat::BuildClusterTree()` and examining pairs of leaf nodes that look ambiguous.
2. Filter down the dataset to contain only the clusters chosen in step 1.
3. Input the filtered data from step 2 as the "target data".
4. Make a decision on whether to merge the clusters by examining the top DE genes discovered by ClusterDE.

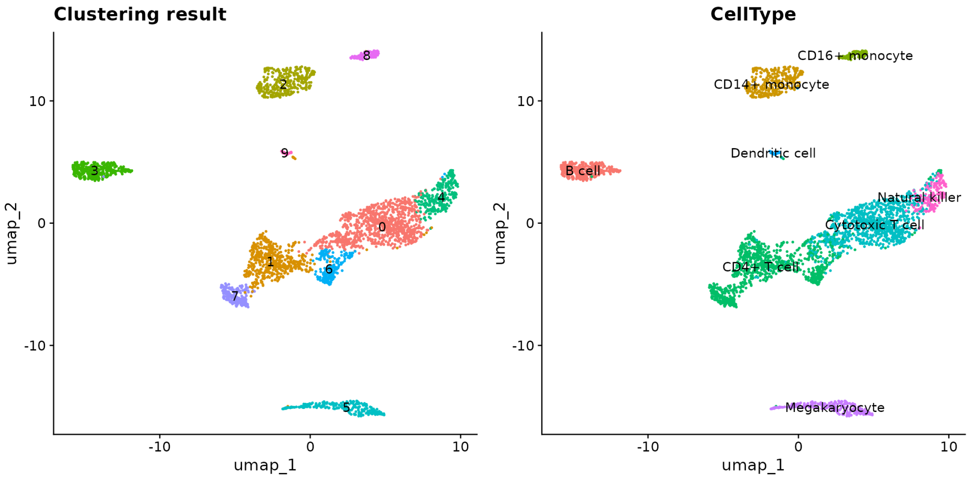


Figure 9: A UMAP plot of a clustered PBMC dataset. We can see that clusters 2 and 8 are close, so they are candidates for input into ClusterDE. Domain knowledge validates this choice, as they represent similar cell types (monocyte subtypes) (Song, Li, and Chen 2024).

How to decide whether to merge clusters

ClusterDE does not perform automatic cluster merging. Its purpose and focus is to identify trustworthy DE genes that can be analyzed downstream to evaluate their appropriateness as marker genes. ClusterDE therefore functions as a tool to help researchers "explore the functional and molecular characteristics of clusters" (Song et al. 2023), not as an automated decision-maker (Song et al. 2023).

Whether you should cluster once or twice

Thus far, we have considered parallel clustering on the target data and synthetic null data. However, this is not strictly necessary. ClusterDE

Choosing a distribution to model the counts

Thus far, we have used the negative binomial distribution to model the gene counts. This is because scRNA-seq data are typically overdispersed, making a Poisson model inappropriate (since in the Poisson distribution, the mean is exactly equal to the variance) (Neufeld et al. 2023).

However, zero-inflated versions of these distributions can also be reasonable choices, since the count data is often very sparse. *BacSC* is a pipeline for bacterial scRNA-seq analysis which defines a protocol for choosing between these four distribution distributions. The sparsity of the count data makes it difficult for optimizers to converge when trying to fit a negative binomial distribution, making the zero-inflated version appropriate in such a situation. We ran into similar issues when trying to generate the synthetic null datasets for this analysis using `ProDG`, an experimental Python package for prokaryotic data generation that uses the copula approach.

Performance of ClusterDE

We discuss some of the benchmarks performed by the ClusterDE authors.

Performance against other DE methods

In one of the authors' benchmarks, they compared ClusterDE against count splitting, the TN test, and the default Seurat pipeline on four PBMC datasets. These datasets were chosen because they were clustered poorly, and therefore good differential expression performance is defined by the discovery of no DE genes (see Figure 11).

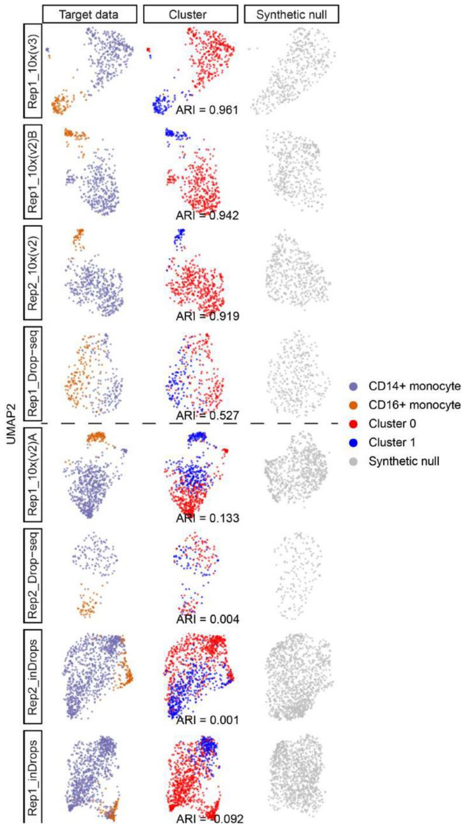


Figure 10: Figure S9 from the ClusterDE paper. 8 PBMC datasets ordered by Seurat clustering accuracy, from highest to lowest. The dashed line divides the more well-clustered datasets (top) from the poorly clustered datasets (bottom) (Song et al. 2023).

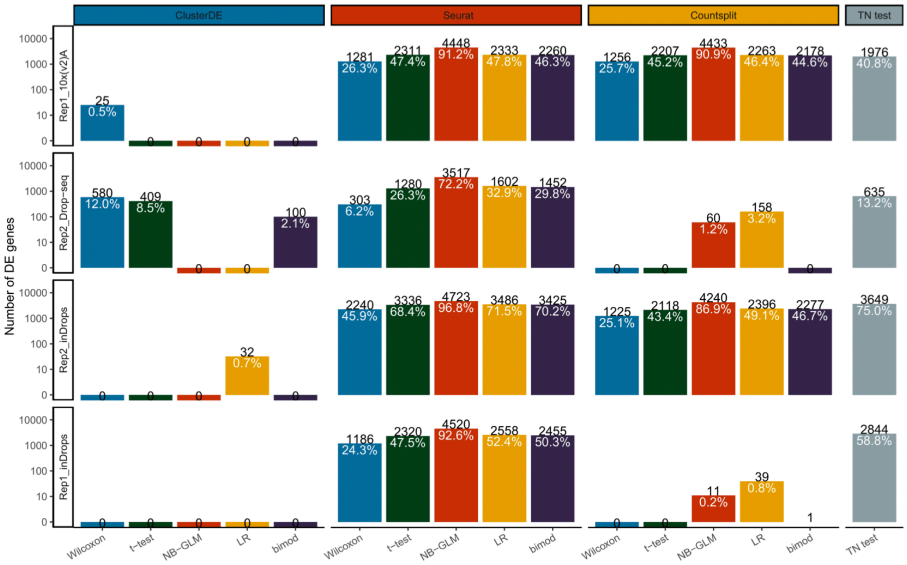


Figure 11: Figure S10 from the ClusterDE paper. No DE genes should be discovered, since the clustering quality is poor. On most of the datasets, and with most of the types of tests, ClusterDE accomplishes this (Song et al. 2023).

Performance against other null generation strategies

Knockoffs

Data analysis

For the seminar, we chose to investigate how a shrinkage estimate of the correlation matrix affects the synthetic null dataset (step 1 of ClusterDE). If the Schaefer-Strimmer shrinkage estimate leads to synthetic null data that more closely matches the original data in terms of summary statistics, then it intuitively a better candidate for DE testing.

B. subtilis 168 dataset

We chose to investigate the *Bsub_minmed_PB* dataset. This is a dataset that was generated by ProBac sequencing (ProBac-seq), in order to validate the performance of this method. ProBac-seq uses messenger RNA-specific probes, and multiple probes per organism, to sequence bacterial samples (McNulty et al. 2023), (Samanta et al. 2024). The *Bsub_minmed_PB* dataset contains the *B. subtilis 168* strain, “grown to late exponential phase in M9 minimal media supplemented with malate” (see Table 1, Ostner et al. 2024), and (McNulty et al. 2023).

This data is analyzed in (Ostner et al. 2024), which proposes *BacSC*, a pipeline for analysis of bacterial scRNA-seq data, which we use and describe below.

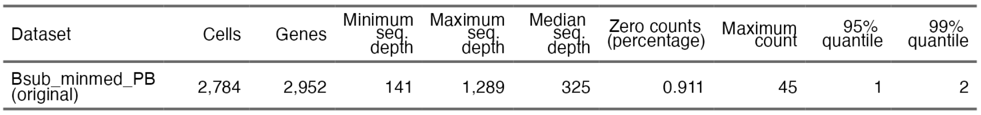
Preprocessing

Because ProBac-seq generates multiple probe reads for each gene, *BacSC* performs **max-pooling**: it takes the maximum count among all probe reads (Goodfellow, Bengio, and Courville 2016).

It filters out all cells with a sequencing depth less than 100: that is, cells with less than 100 genes expressed. Furthermore, it filters out genes present in only 1 cell.

Note that mitochondrial genes are not filtered out here, while they would be for eukaryotic scRNA-seq data. This is because bacteria do not have mitochondria (Ostner et al. 2024).

See Figure 12 for a summary of the dataset, and Ostner et al. for a more detailed description (Ostner et al. 2024).



Dataset	Cells	Genes	Minimum seq. depth	Maximum seq. depth	Median seq. depth	Zero counts (percentage)	Maximum count	95% quantile	99% quantile
Bsub_minmed_PB (original)	2,784	2,952	141	1,289	325	0.911	45	1	2

Figure 12: Summary statistics for the *Bsub_minmed_PB* after quality control.

Synthetic null data generation

Recall that the purpose of this analysis is to investigate the differences between empirical correlation estimates for null data generation, which have existing straightforward implementations, and the Schaefer-Strimmer shrinkage estimate.

Schaefer-Strimmer estimation of the correlation matrix

Badri et al. showed that the Schaefer-Strimmer shrinkage estimate Badri et al. (2020). Furthermore, the sample correlation matrix is inadmissible for high-dimensional data, which is the typical setting for bacterial scRNA-seq analysis Ostner et al. (2024). Thus, there is a theoretical and empirical justification for using a shrinkage estimate of the correlation matrix.

Ostner provided a Python implementation of the following estimator proposed by Badri et al. (Badri et al. 2020).

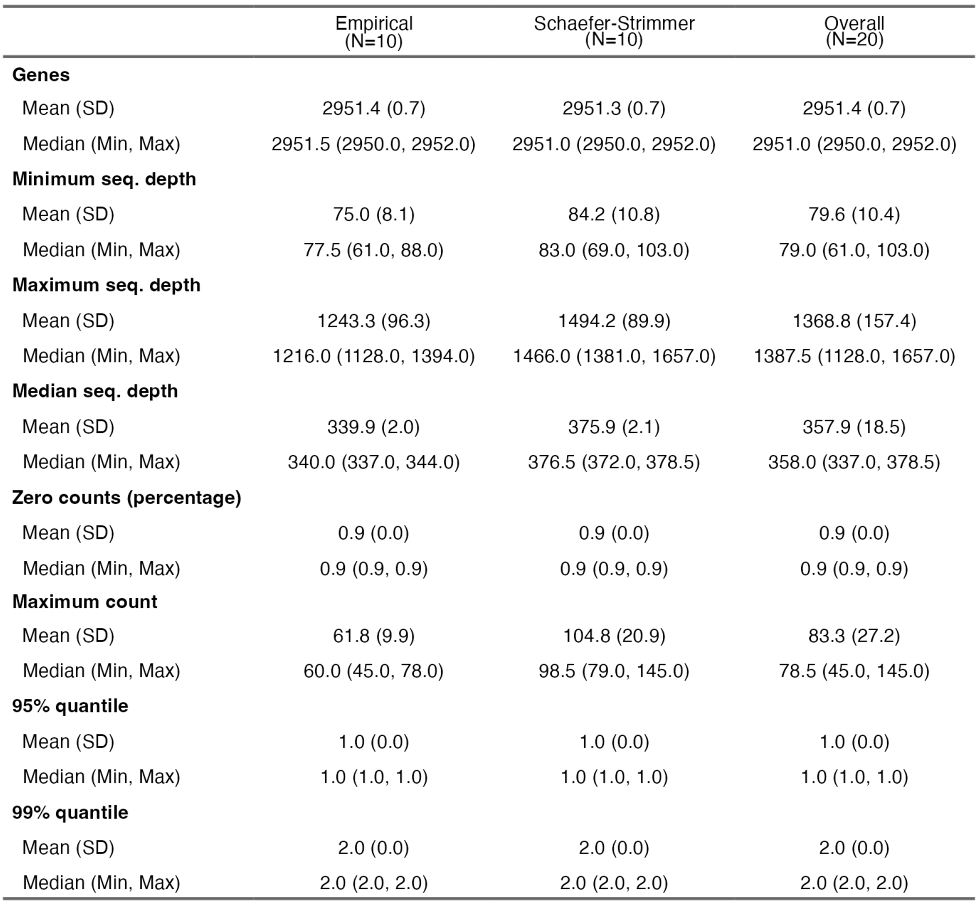
Let S^* be the shrinkage covariance matrix. The covariances (off-diagonal entries) and variances (diagonal entries) are shrunk separately.

The off-diagonal entries of S^* are defined as:

$$s^*_{ik} = r$$

Results

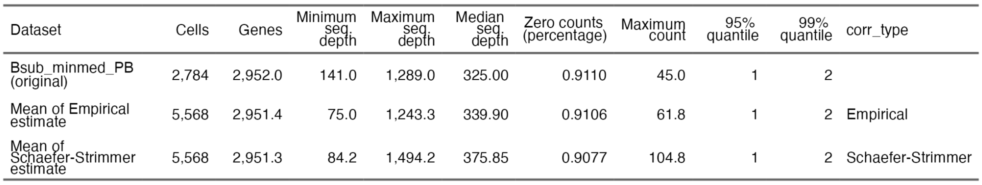
Generally, the variance of each summary statistic looks subjectively similar across the two estimation procedures. The 95% and 99% quantiles of the counts are always the same.



	Empirical (N=10)	Schaefer-Strimmer (N=10)	Overall (N=20)
Genes			
Mean (SD)	2951.4 (0.7)	2951.3 (0.7)	2951.4 (0.7)
Median (Min, Max)	2951.5 (2950.0, 2952.0)	2951.0 (2950.0, 2952.0)	2951.0 (2950.0, 2952.0)
Minimum seq. depth			
Mean (SD)	75.0 (8.1)	84.2 (10.8)	79.6 (10.4)
Median (Min, Max)	77.5 (61.0, 88.0)	83.0 (69.0, 103.0)	79.0 (61.0, 103.0)
Maximum seq. depth			
Mean (SD)	1243.3 (96.3)	1494.2 (89.9)	1368.8 (157.4)
Median (Min, Max)	1216.0 (1128.0, 1394.0)	1466.0 (1381.0, 1657.0)	1387.5 (1128.0, 1657.0)
Median seq. depth			
Mean (SD)	339.9 (2.0)	375.9 (2.1)	357.9 (18.5)
Median (Min, Max)	340.0 (337.0, 344.0)	376.5 (372.0, 378.5)	358.0 (337.0, 378.5)
Zero counts (percentage)			
Mean (SD)	0.9 (0.0)	0.9 (0.0)	0.9 (0.0)
Median (Min, Max)	0.9 (0.9, 0.9)	0.9 (0.9, 0.9)	0.9 (0.9, 0.9)
Maximum count			
Mean (SD)	61.8 (9.9)	104.8 (20.9)	83.3 (27.2)
Median (Min, Max)	60.0 (45.0, 78.0)	98.5 (79.0, 145.0)	78.5 (45.0, 145.0)
95% quantile			
Mean (SD)	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
Median (Min, Max)	1.0 (1.0, 1.0)	1.0 (1.0, 1.0)	1.0 (1.0, 1.0)
99% quantile			
Mean (SD)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)
Median (Min, Max)	2.0 (2.0, 2.0)	2.0 (2.0, 2.0)	2.0 (2.0, 2.0)

Figure 13: Summary statistics comparing the two correlation estimation strategies investigated in this analysis.

Figure 14 displays the mean summary statistic for each estimation strategy along with the original data. From this table, we can see that the Schaefer-Strimmer estimates yield synthetic null datasets with systematically larger counts, and larger sequencing depths. The empirical correlation estimates match the original data better in terms of median sequencing depth, maximum sequencing depth, percentage of zero counts, and maximum count.



Dataset	Cells	Genes	Minimum seq. depth	Maximum seq. depth	Median seq. depth	Zero counts (percentage)	Maximum count	95% quantile	99% quantile	corr_type
Bsub_minmed_PB (original)	2,784	2,952.0	141.0	1,289.0	325.00	0.9110	45.0	1	2	
Mean of Empirical estimate	5,568	2,951.4	75.0	1,243.3	339.90	0.9106	61.8	1	2	Empirical
Mean of Schaefer-Strimmer estimate	5,568	2,951.3	84.2	1,494.2	375.85	0.9077	104.8	1	2	Schaefer-Strimmer

Figure 14: Summary statistics comparing the two correlation estimation strategies investigated in this analysis.

Discussion

In terms of the summary statistics examined here, the two correlation estimators do not look substantially different. However, one surprising result is that the Schaefer-Strimmer datasets have systematically larger maximum counts, where the minimum maximum count for a Schaefer-Strimmer dataset is higher than the maximum maximum count for an empirical correlation dataset. This is surprising because the Schaefer-Strimmer estimate is a shrinkage estimate, and we would expect that if the variances of the count distributions have been shrunk, then we would observe extremely large counts less often. Further exploration is required to determine the cause of this phenomenon.

No differential expression testing was conducted here, so the sensitivity of the actual DE genes outputted by ClusterDE to the correlation estimator is currently unknown. Based on these initial results, we do not expect a substantial difference in the quality of DE genes outputted by ClusterDE.

Further investigations into ClusterDE could include a benchmark of different copula families. We only considered the Gaussian copula here. The ClusterDE package also supports vine copulas, which they say are better but computationally very expensive. A more detailed investigation of the computation vs. performance tradeoff could be valuable here.

Furthermore, *BacSC* implements some extensions to the ClusterDE method.

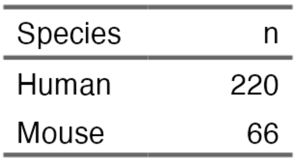
Conclusion

Cell annotation provides analysts and domain experts with important information about their scRNA-seq samples. While various forms of annotation exist, including completely automated annotation, one of the most popular strategies for annotation is manual annotation based on a set of marker genes. To discover new marker genes, analysts often perform differential expression testing, in which they first cluster their cells into two groups, and then test each gene for differences between the groups. However, this process double-dips, and therefore leads to an inflated false discovery rate if executed naively. Various methods exist to counteract this, but ClusterDE performs the best on benchmarks. ClusterDE's use of a synthetic null dataset as a negative control, combined with its choice of cutoff using Clipper, allow it to successfully control the false discovery rate.

In this paper, a shrinkage estimate of the covariance matrix was explored as a potential extension to ClusterDE's null data generation step. Despite previous theoretical and empirical justifications for its use, this Schaefer-Strimmer covariance estimate does not produce data that matches the original data noticeably better than an empirical correlation estimate.

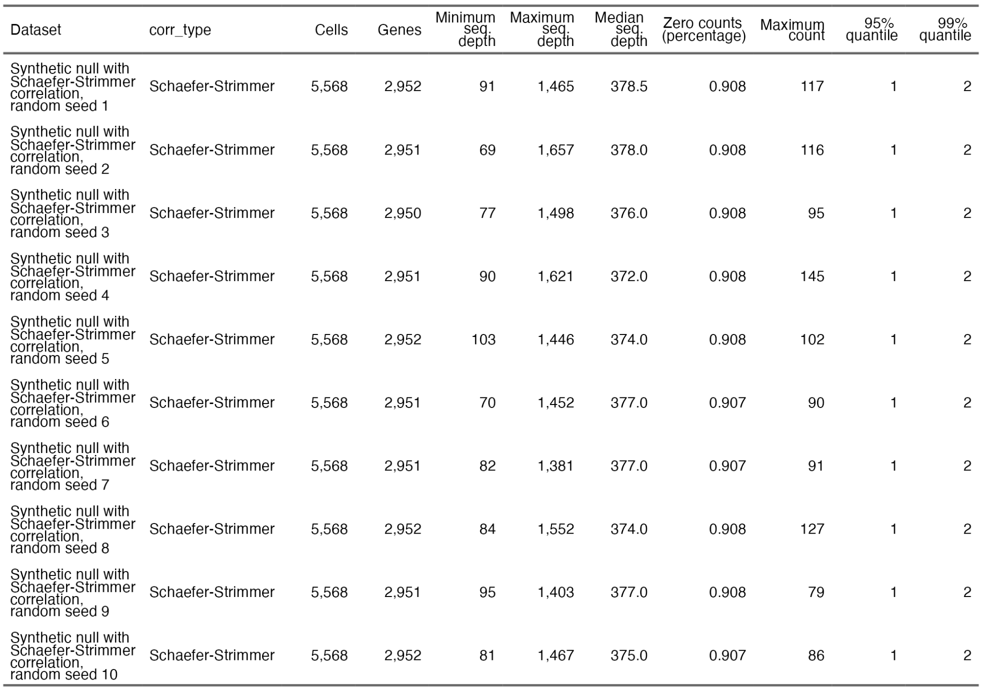
Appendix

The code used to generate this paper and perform this analysis is available at this Github repository.



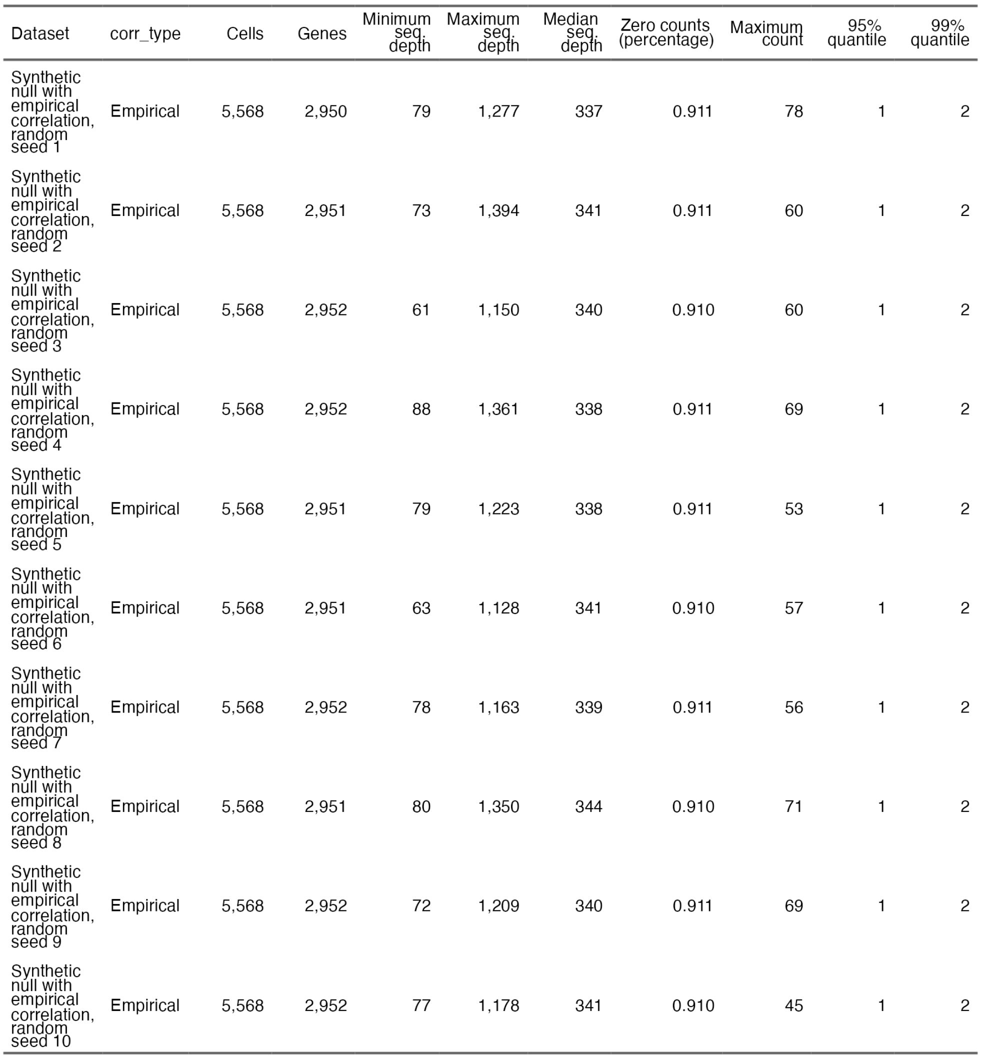
Species	n
Human	220
Mouse	66

Figure 15: CD19 species counts.



Dataset	corr_type	Cells	Genes	Minimum seq. depth	Maximum seq. depth	Median seq. depth	Zero counts (percentage)	Maximum count	95% quantile	99% quantile
Synthetic null with Schaefer-Strimmer correlation, random seed 1	Schaefer-Strimmer	5,568	2,952	91	1,465	378.5	0.908	117	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 2	Schaefer-Strimmer	5,568	2,951	69	1,657	378.0	0.908	116	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 3	Schaefer-Strimmer	5,568	2,950	77	1,498	376.0	0.908	95	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 4	Schaefer-Strimmer	5,568	2,951	90	1,621	372.0	0.908	145	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 5	Schaefer-Strimmer	5,568	2,952	103	1,446	374.0	0.908	102	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 6	Schaefer-Strimmer	5,568	2,951	70	1,452	377.0	0.907	90	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 7	Schaefer-Strimmer	5,568	2,951	82	1,381	377.0	0.907	91	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 8	Schaefer-Strimmer	5,568	2,952	84	1,552	374.0	0.908	127	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 9	Schaefer-Strimmer	5,568	2,951	95	1,403	377.0	0.908	79	1	2
Synthetic null with Schaefer-Strimmer correlation, random seed 10	Schaefer-Strimmer	5,568	2,952	81	1,467	375.0	0.907	86	1	2

Figure 16: Summary statistics for all synthetic datasets generated with Schaefer-Strimmer correlation estimates.



Dataset	corr_type	Cells	Genes	Minimum seq. depth	Maximum seq. depth	Median seq. depth	Zero counts (percentage)	Maximum count	95% quantile	99% quantile
Synthetic null with empirical correlation, random seed 1	Empirical	5,568	2,950	79	1,277	337	0.911	78	1	2
Synthetic null with empirical correlation, random seed 2	Empirical	5,568	2,951	73	1,394	341	0.911	60	1	2
Synthetic null with empirical correlation, random seed 3	Empirical	5,568	2,952	61	1,150	340	0.910	60	1	2
Synthetic null with empirical correlation, random seed 4	Empirical	5,568	2,952	88	1,361	338	0.911	69	1	2
Synthetic null with empirical correlation, random seed 5	Empirical	5,568	2,951	79	1,223	338	0.911	53	1	2
Synthetic null with empirical correlation, random seed 6	Empirical	5,568	2,951	63	1,128	341	0.910	57	1	2
Synthetic null with empirical correlation, random seed 7	Empirical	5,568	2,952	78	1,163	339	0.911	56	1	2
Synthetic null with empirical correlation, random seed 8	Empirical	5,568	2,951	80	1,350	344	0.910	71	1	2
Synthetic null with empirical correlation, random seed 9	Empirical	5,568	2,952	72	1,209	340	0.911	69	1	2
Synthetic null with empirical correlation, random seed 10	Empirical	5,568	2,952	77	1,178	341	0.910	45	1	2

Figure 17: Summary statistics for all datasets generated with empirical correlation estimates.

References

Badri, Michelle, Zachary D Kurtz, Richard Bonneau, and Christian L Müller. 2020. "Shrinkage improves estimation of microbial associations under different normalization methods." *NAR Genomics and Bioinformatics* 2 (4): lqaa100. https://doi.org/10.1093/nargab/lqaa100.

Casella, George, and Roger Berger. 2001. *Statistical Inference*. Textbook Binding; Duxbury Resource Center.

Czado, Claudia. 2019. *Analyzing Dependent Data with Vine Copulas: A Practical Guide with r*. First. Springer Cham.

Ekins, S, J Mestres, and B Testa. 2007. "In Silico Pharmacology for Drug Discovery: Methods for Virtual Ligand Screening and Profiling." *British Journal of Pharmacology* 152 (1): 9–20. https://doi.org/https://doi.org/10.1038/sj.bjp.0707305.

Ge, Xinzhou, Yiling Elaine Chen, Dongyuan Song, MeiLu McDermott, Kyla Woyshner, Antigoni Manousopoulou, Ning Wang, Wei Li, Leo D. Wang, and Jingyi Jessica Li. 2021. "Clipper: P-Value-Free FDR Control on High-Throughput Data from Two Conditions." *Genome Biology* 22 (1): 288. https://doi.org/10.1186/s13059-021-02506-9.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Green, Eric, Lawrence Brody, Sarah Bates, Mary Beth Gardiner, Jill Thomas, Britny Kish, Darryl Leja, et al. 2024. "Lymphocyte." https://www.genome.gov/genetics-glossary/Lymphocyte.

Hao, Yuhan, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck III, Shiwei Zheng, Andrew Butler, Maddie J. Lee, et al. 2021. "Integrated Analysis of Multimodal Single-Cell Data." *Cell*. https://doi.org/10.1016/j.cell.2021.04.048.

Heumos, Lukas, Anna C. Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D. Lücken, et al. 2023. "Best Practices for Single-Cell Analysis Across Modalities." *Nature Reviews Genetics* 24 (8): 550–72. https://doi.org/10.1038/s41576-023-00586-w.

Hu, Congxue, Tengyue Li, Yingqi Xu, Xinxin Zhang, Feng Li, Jing Bai, Jing Chen, et al. 2022. "CellMarker 2.0: an updated database of manually curated cell markers in human/mouse and web tools based on scRNA-seq data." *Nucleic Acids Research* 51 (D1): D870–76. https://doi.org/10.1093/nar/gkac947.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2021. *An Introduction to Statistical Learning with Applications in r*. Second. Springer Texts in Statistics.

McNulty, Ryan, Duluxan Sritharan, Seong Ho Pahng, Jeffrey P. Meisch, Shichen Liu, Melanie A. Brennan, Gerda Saxer, Sahand Hormoz, and Adam Z. Rosenthal. 2023. "Probe-Based Bacterial Single-Cell RNA Sequencing Predicts Toxin Regulation." *Nature Microbiology* 8 (5): 934–45. https://doi.org/10.1038/s41564-023-01348-4.

Nelsen, Roger B. 2006. *An Introduction to Copulas*. First. Springer New York, NY.

Neufeld, Anna, Joshua Popp, Lucy L. Gao, Alexis Battle, and Daniela Witten. 2023. "Negative Binomial Count Splitting for Single-Cell RNA Sequencing Data." https://arxiv.org/abs/2307.12985.

Ostner, Johannes, Tim Kirk, Roberto Olayo-Alarcon, Janne Gesine Thöming, Adam Z. Rosenthal, Susanne Häussler, and Christian L. Müller. 2024. "BacSC: A General Workflow for Bacterial Single-Cell RNA Sequencing Data Analysis." *bioRxiv*. https://doi.org/10.1101/2024.06.22.600071.

"Probability Integral Transform." n.d. https://en.wikipedia.org/wiki/Probability_integral_transform.

Rafi, Zad, and Sander Greenland. 2020. "Semantic and Cognitive Tools to Aid Statistical Science: Replace Confidence and Significance by Compatibility and Surprise." *BMC Medical Research Methodology* 20 (1): 244. https://doi.org/10.1186/s12874-020-01105-9.

Samanta, Prosenjit, Samuel F. Cooke, Ryan McNulty, Sahand Hormoz, and Adam Rosenthal. 2024. "ProBac-Seq, a Bacterial Single-Cell RNA Sequencing Methodology Using Droplet Microfluidics and Large Oligonucleotide Probe Sets." *Nature Protocols*. https://doi.org/10.1038/s41596-024-01002-1.

Song, Dongyuan, Kexin Li, and Siqi Chen. 2024. *ClusterDE: A Post-Clustering Differential Expression (DE) Method for Solving Double Dipping*. https://github.com/SONGDONGYUAN1994/ClusterDE.

Song, Dongyuan, Kexin Li, Xinzhou Ge, and Jingyi Jessica Li. 2023. "ClusterDE: A Post-Clustering Differential Expression (DE) Method Robust to False-Positive Inflation Caused by Double Dipping." *bioRxiv*. https://doi.org/10.1101/2023.07.21.550107.

Song, Dongyuan, Qingyang Wang, Guanao Yan, Tianyang Liu, Tianyi Sun, and Jingyi Jessica Li. 2024. "scDesign3 Generates Realistic in Silico Data for Multimodal Single-Cell and Spatial Omics." *Nature Biotechnology* 42 (2): 247–52. https://doi.org/10.1038/s41587-023-01772-1.

Traag, V. A., L. Waltman, and N. J. van Eck. 2019. "From Louvain to Leiden: Guaranteeing Well-Connected Communities." *Scientific Reports* 9 (1): 5233. https://doi.org/10.1038/s41598-019-41695-z.

Zhang, Carson. 2023. "Probability Integral Transform." https://blog.carsonzhang.com/posts/probability_integral_transform/.