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. (2023) propose ClusterDE, 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.

TODO: distinction between different types of T cells requires better annotation methods.

TODO: 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.

These marker genes are compiled into databases such as CellMarker 2.0, a database containing (Hu et al. 2022).

(1/6)

https://academic.oup.com/nar/article/51/D1/D870/6775381

http://bio-bigdata.hrbmu.edu.cn/CellMarker/

https://www.nature.com/articles/s41467-022-28803-w

Automated annotation

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.

(define validity)

TODO: mention Scanpy, Seurat, and their default methods

To identify these cell types, they identify a set of cell-type marker genes.

To identify these cell-type marker genes, they perform differential expression testing.

Naive differential expression testing is susceptible to false discoveries caused by double-dipping.

The double-dipping issue

Mention Scanpy, Seurat defaults

Mention warnings but lack of solutions

Toy example illustrating double dipping

False discoveries

Consider the notation defined in Figure 1.

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 (James et al. 2021), which we call the **false discovery rate** $E[\frac{V}{R}]$.

	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 1: 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 $\times$ 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

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

.

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

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

.

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

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

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 2.

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).
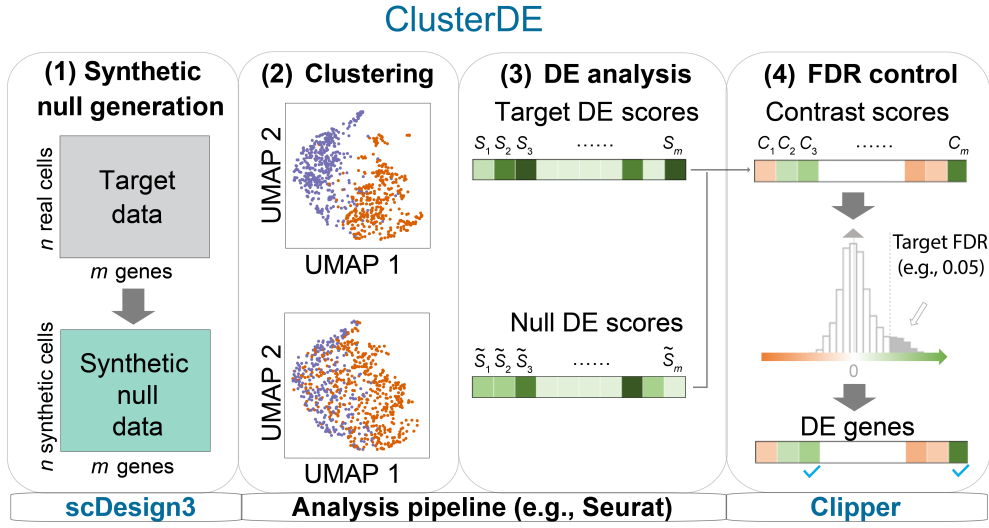
Figure 2: A visual overview of the ClusterDE method. In step 1, a negative control dataset is generated. In step 2, a clustering algorithm is applied to each dataset. In step 3, a differential expression test is performed for each gene, computing a DE score for each gene in each dataset. In step 4, the 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:

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

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 **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 4).
- 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 3).
- 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.

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

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
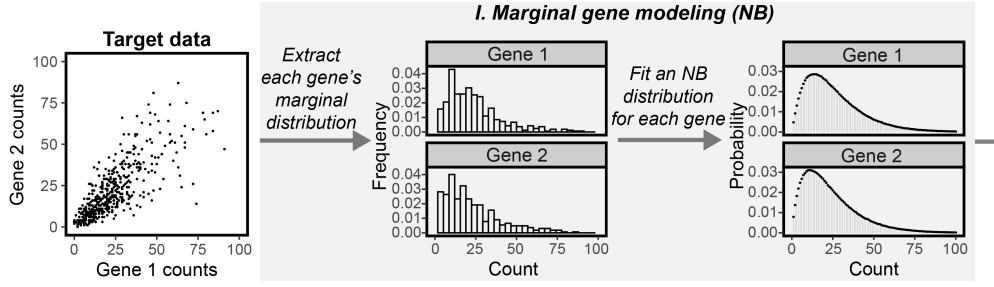
Figure 3: Fit a marginal distribution for each gene. The copula approach allows us to model the marginal distributions separately from the covariance structure of the variables (see Figure 4) (Song, Li, and Chen 2024).
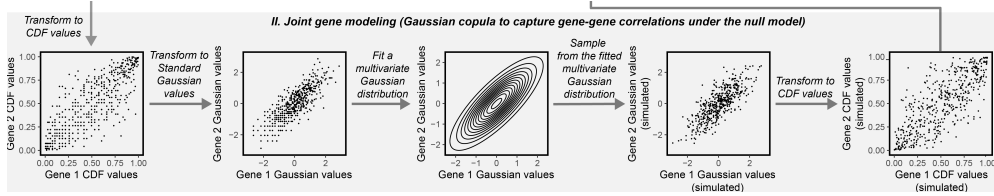
Figure 4: Estimate the covariance matrix for the m-dimensional gene distribution. The copula approach allows us to model the correlations between genes separately from their marginal distributions (see Figure 3) (Song, Li, and Chen 2024).

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).

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.
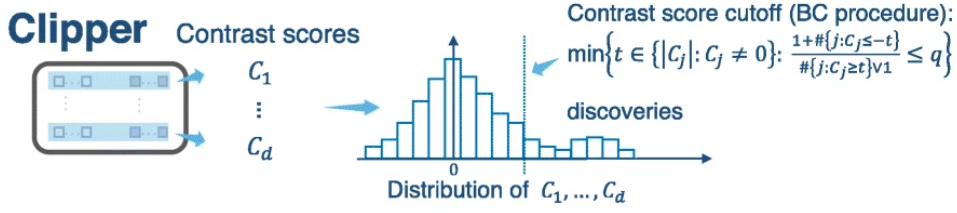
Figure 5: The Clipper method for FDR control. Part of Figure 1.b. from Ge et al. (Ge et al. 2021).

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 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 6). 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.
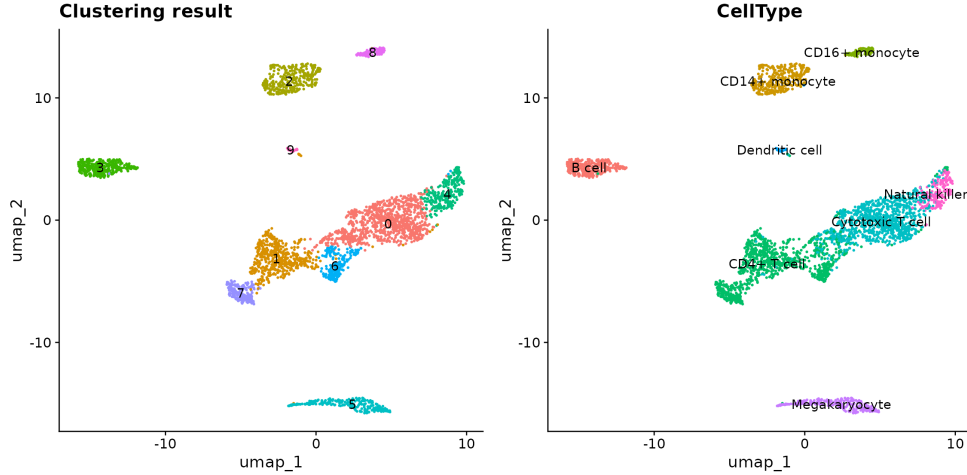
Figure 6: A UMAP plot of a clustered PBMC dataset. We can see that clusters 2 and 8 are close, so they are candidates for input into ClusterDE. Domain knowledge validates this choice, as they represent similar cell types (monocyte subtypes) (Song, Li, and Chen 2024).

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 verse sparse.

We defer a more detailed discussion to later, where we describe *BacSC*: a pipeline for bacterial scRNA-seq analysis which defines a protocol for choosing a distribution.

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 8).
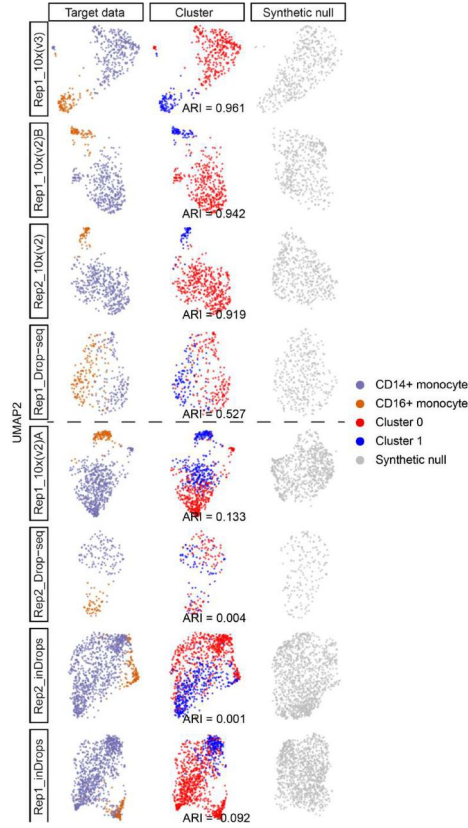
Figure 7: Figure S9 from the ClusterDE paper. 8 PBMC datasets ordered by Seurat clustering accuracy, from highest to lowest. The dashed line divides the more well-clustered datasets (top) from the poorly clustered datasets (bottom) (Song et al. 2023).

Performance against other null generation strategies

Data analysis

For the seminar, we chose to investigate how an shrinkage esimate of the covariance matrix affects the synthetic null dataset (step 1 of ClusterDE).

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.

14
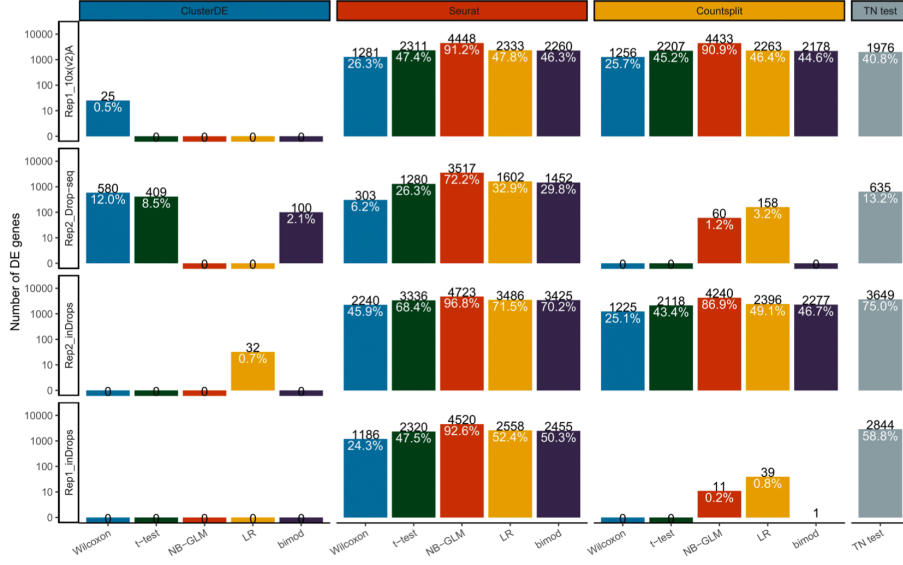
Figure 8: Figure S10 from the ClusterDE paper. No DE genes should be discovered, since the clustering quality is poor. On most of the datasets, and with most of the types of tests, ClusterDE accomplishes this (Song et al. 2023).

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).

Synthetic null data generation

Schäfer-Strimmer

Badri et al. (2020) showed that shrinkage

Results

```
# library(flextable)
# library(tinytable)
# library(readr)
```

```
# table_e1 = read_csv("../seminar_paper-bacsc/python/reproduce_results/table_e1.csv")
# flextable(table_e1)
#
# table_e1_synthetic = read_csv("../seminar_paper-bacsc/python/synthetic_null_generation/table_
# flextable(table_e1_synthetic)
```

Simulation study

Discussion

Different copula generation strategies.

Appendix

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.

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/.