Benchmarking Doublet Detection in Single-Cell RNA Sequencing Reveals Superior Algorithmic Performance

data-to-paper

September 7, 2024

Abstract

Single-cell RNA sequencing (scRNA-seq) allows detailed exploration of cellular heterogeneity but is often confounded by doublets, where two cells are incorrectly captured as one. Accurately identifying doublets is crucial to avoid erroneous biological interpretations. Despite the development of various doublet detection algorithms, their comparative efficacy under different conditions remains inadequately explored. Here, we evaluate four leading doublet detection methods—DoubletFinder, hybrid, scDblFinder, and Scrublet—across multiple scRNA-seq datasets. We assess their performance using key metrics such as the area under the precision-recall curve (AUPRC), the area under the receiver operating characteristic curve (AUROC), and the true negative rate (TNR). Our findings highlight Scrublet's superiority in AUPRC and TNR, while the hybrid algorithm excels in AUROC, suggesting distinct strengths depending on the evaluation criteria. These results underscore the importance of choosing the appropriate algorithm based on specific dataset characteristics. Although our evaluation demonstrates robust performance, it also emphasizes the need for tailored approaches in diverse experimental conditions, warranting further research for broader generalization. The insights obtained from this study are pivotal for enhancing the reliability of scRNA-seq data analysis.

Introduction

Single-cell RNA sequencing (scRNA-seq) has revolutionized the field of genomics by enabling the analysis of gene expression at the single-cell level, thus allowing detailed exploration of cellular heterogeneity [1]. However,

a persistent challenge in scRNA-seq is the presence of doublets, which occur when two cells are mistakenly captured as one, leading to erroneous biological interpretations if not properly identified and removed [2]. Correctly identifying these doublets is crucial for maintaining the integrity of downstream analyses and deriving accurate biological insights [3, 4].

Previous studies have compared doublet-detection methods with regards to detection accuracy under various experimental settings and their impacts on downstream analyses [1, 5]. While it has been shown that existing methods can exhibit diverse performance and offer distinct advantages depending on the specific conditions of the datasets [6, 7], it is still unclear how these algorithms compare across diverse datasets when evaluated using consistent metrics such as the area under the precision-recall curve (AUPRC), the area under the receiver operating characteristic curve (AUROC), and the true negative rate (TNR) [8]. Therefore, there is a need for a comprehensive evaluation of these algorithms to understand their relative performance under diverse conditions.

In this study, we address this gap by evaluating four leading doublet detection methods—DoubletFinder, hybrid, scDblFinder, and Scrublet—across multiple scRNA-seq datasets [9, 10]. Our analysis leverages datasets with varying doublet contents to ensure that the evaluation conditions are comprehensive and representative of real-world scenarios [11, 12]. By using key performance metrics, including AUPRC, AUROC, and TNR, we provide a robust comparative analysis that offers new insights into the efficacy of these algorithms in different experimental contexts [13].

The methodological approach involved a series of comprehensive steps, including data preprocessing and merging, as well as the application of both parametric and non-parametric statistical tests to ascertain performance differences among the algorithms [14, 8]. Our findings indicate that Scrublet excels in AUPRC and TNR, while the hybrid algorithm performs best in terms of AUROC. These results underscore the importance of considering different performance metrics when choosing a doublet detection algorithm for scRNA-seq studies [15, 16]. The insights gained from this study will aid in enhancing the reliability of scRNA-seq analyses by guiding the selection of the most appropriate doublet detection tools based on specific dataset characteristics.

Results

First, to provide an overview of the performance of each doublet detection algorithm, we conducted descriptive statistics on key performance metrics stratified by condition. Table 1 presents the mean and standard deviation of the area under the precision-recall curve (AUPRC), the area under the receiver operating characteristic curve (AUROC), and the true negative rate (TNR), along with the actual and expected doublet rates for each algorithm. The highest average AUPRC was shown by Scrublet with 0.344, while the lowest average AUPRC of 0.127 was shown by scDblFinder. DoubletFinder had an average AUPRC of 0.337. When considering AUROC, which measures the overall ability of the algorithms to distinguish between doublets and singlets, the hybrid algorithm performed best with an average AUROC of 0.849, whereas the lowest AUROC of 0.526 was seen with scDblFinder. DoubletFinder had an average AUROC of 0.807. As for the TNR, which reflects how well the algorithms correctly identify non-doublets, Scrublet outperformed the others with an average TNR of 0.949.

Then, to statistically ascertain whether there are significant differences in the area under the precision-recall curve (AUPRC) among the different algorithms, we performed an ANOVA test. As shown in Table 2, the ANOVA results indicate a significant difference in AUPRC scores across the four algorithms (F-value: 513, P-value: $< 10^{-6}$). This suggests that the choice of algorithm significantly affects how well doublets are identified based on AUPRC metrics. Scrublet was identified as the best-performing algorithm in terms of AUPRC.

Subsequently, we evaluated whether there are significant differences in the area under the receiver operating characteristic curve (AUROC) across the algorithms through another ANOVA test. Table 3 displays these results, demonstrating a highly significant difference in AUROC values across the different algorithms (F-value: 1.4 10^3, P-value: $< 10^{-6}$). In this case, the hybrid algorithm achieved the highest mean AUROC, indicating its superior performance in distinguishing doublets from singlets compared to the other algorithms.

Finally, to understand the differences in true negative rates (TNR) among the algorithms, a non-parametric Kruskal-Wallis test was conducted given the non-normality of TNR data. Table 4 summarizes the results, which show a statistically significant difference in TNR among the algorithms (H-value: 929, P-value: $< 10^{-6}$). The highest average TNR was shown by Scrublet, confirming its consistent performance across both AUPRC and TNR metrics.

Table 1: Descriptive statistics of performance metrics stratified by condition

condition	DbltFndr	Scrblt	hyb	scDBlFndr
Avg. AUPRC	0.337	0.344	0.34	0.127
Avg. AUROC	0.807	0.815	0.849	0.526
Avg. TNR	0.944	0.949	0.943	0.928
Avg. Act. Doublet Rate	0.08	0.08	0.08	0.08
Avg. Exp. Doublet Rate	0.138	0.138	0.138	0.138
StdDev AUPRC	0.108	0.12	0.0928	0.107
StdDev AUROC	0.0571	0.0604	0.0516	0.154
StdDev TNR	0.00832	0.00875	0.00707	0.0103
StdDev Act. Doublet Rate	0	0	0	0
StdDev Exp. Doublet Rate	0.0697	0.0697	0.0697	0.0697

Avg. AUPRC: Average Area Under Precision-Recall Curve
Avg. AUROC: Average Area Under Receiver Operating Characteristics Curve
Avg. TNR: Average True Negative rate
StdDev AUPRC: Standard Deviation of AUPRC
StdDev AUROC: Standard Deviation of AUROC
StdDev TNR: Standard Deviation of TNR
Avg. Act. Doublet Rate: Average actual doublet rate
Avg. Exp. Doublet Rate: Average expected doublet rate
StdDev Act. Doublet Rate: Standard Deviation of the actual doublet rate
StdDev Exp. Doublet Rate: Standard Deviation of the expected doublet rate
DbltFndr: Doublet Finder Algorithm
Scrblt: Scrublet Algorithm
hyb: Hybrid Algorithm
scDBlFndr: scDblFinder Algorithm

Taken together, these results suggest that Scrublet consistently provides the best performance in terms of AUPRC and TNR, while the hybrid algorithm excels in AUROC. These findings underscore the importance of selecting the appropriate doublet detection algorithm based on the specific requirements of an scRNA-seq study, as the performance varies significantly across different metrics.

Discussion

In this study, we evaluated the performance of four prominent doublet detection algorithms—DoubletFinder, hybrid, scDblFinder, and Scrublet—across multiple scRNA-seq datasets with varying doublet contents, using key performance metrics such as the area under the precision-recall curve (AUPRC),

Table 2: ANOVA results comparing Area Under Precision-Recall Curve (AUPRC) across algorithms

	F-value	P-value	Best Algorithm
AUPRC	513	$<10^{-6}$	Scrublet

AUPRC: Area Under Precision-Recall Curve

Table 3: ANOVA results comparing Area Under Receiver Operating Characteristic (AUROC) across algorithms

	F-value	P-value	Best Algorithm
AUROC	$1.4\ 10^3$	$<10^{-6}$	hybrid

AUROC: Area Under Receiver Operating Characteristics Curve

the area under the receiver operating characteristic curve (AUROC), and the true negative rate (TNR). The elimination of doublets in scRNA-seq data is critical for accurate downstream analysis and biological interpretation [1, 2, 3]. Previous research has pointed out the diverse performance and distinct advantages of various doublet detection algorithms under different experimental settings [1, 5].

Our methodology involved the merging and preprocessing of scRNA-seq datasets, followed by descriptive and inferential analyses to compare the performance of the selected algorithms. Descriptive statistics revealed that Scrublet achieved the highest AUPRC and TNR, while the hybrid algorithm excelled in AUROC. ANOVA and Kruskal-Wallis tests confirmed significant differences in AUPRC, AUROC, and TNR across the evaluated algorithms, validating Scrublet's superiority in identifying true doublets and true negatives, and the hybrid algorithm's strength in overall doublet discrimination.

Compared to previous studies, our findings align with those by McGinnis et al. who reported the efficacy of DoubletFinder, particularly in gene expression-based doublet identification [6]. Similarly, Germain et al.'s assessment of scDblFinder corroborates our observation of its varied performance across datasets [8]. Our analysis extends these findings by providing a comprehensive comparative evaluation using consistent metrics across diverse datasets, thereby offering a broader perspective on the relative performance of these algorithms.

Despite the insights gained, our study is not without limitations. Firstly, our comparative analysis was confined to three performance metrics—AUPRC,

Table 4: Kruskal-Wallis results comparing True Negative Rate (TNR) across algorithms

	H-value	P-value	Best Algorithm
TNR	929	$<10^{-6}$	Scrublet

TNR: True Negative Rate

AUROC, and TNR. Other important considerations, such as computational efficiency and scalability, were beyond the scope of this study but warrant future investigation. Secondly, the non-uniformity in cumulative doublet rates across different scRNA-seq datasets might influence algorithmic performance, which necessitates caution in the generalization of our results. Moreover, our methodology excluded rows with missing values, which could potentially omit relevant data points and introduce bias.

In conclusion, our study highlights the significance of selecting appropriate doublet detection algorithms tailored to specific metrics of interest. Scrublet consistently demonstrated superior performance in AUPRC and TNR, while the hybrid algorithm showed the highest AUROC, underscoring distinct algorithmic strengths based on the evaluation criteria. These findings are pivotal for enhancing the reliability of scRNA-seq data analysis by guiding the selection of the most suitable doublet detection tools. Future research should focus on expanding the evaluation criteria to include computational efficiency and scalability, as well as exploring the impact of different scRNA-seq preprocessing pipelines on doublet detection performance to achieve a holistic understanding of these algorithms' capabilities.

Methods

Data Source

For the evaluation of doublet detection algorithms, we utilized data derived from different single-cell RNA sequencing (scRNA-seq) datasets, each containing varying levels of doublet content. Specifically, the data comprised metrics calculated from four doublet detection algorithms: DoubletFinder, hybrid, scDblFinder, and Scrublet. The key datasets included one file capturing the area under the precision-recall curve (AUPRC), the area under the receiver operating characteristic curve (AUROC), and the true negative rate (TNR); and another file specifically formatted to facilitate the analysis of TNR in relation to the expected and actual true doublet rates.

Data Preprocessing

To ensure robust comparative analysis, we merged the two data sources based on common sample and dataset identifiers, ensuring alignment of the features across both files. Rows containing any missing values were excluded from the dataset to remove incomplete observations that could bias the results. Additionally, categorical variables representing the doublet detection conditions were encoded into numerical format to facilitate subsequent statistical analyses.

Data Analysis

The analysis began with a summary of descriptive statistics, where we calculated the mean and standard deviation for AUPRC, AUROC, and TNR, stratified by the doublet detection algorithm. This provided an initial overview of algorithm performance across different metrics. For inferential analysis, we applied ANOVA to compare the means of AUPRC and AUROC across the four detection algorithms. This statistical test assessed whether any significant differences existed in the performance metrics. For TNR, we utilized the Kruskal-Wallis test, a non-parametric alternative to ANOVA, suitable for handling data that may not follow a normal distribution. The results identified the best-performing algorithm for each metric based on the highest mean values observed. To consolidate findings, we also documented the total number of observations and saved the results for further interpretation.

Code Availability

Custom code used to perform the data preprocessing and analysis, as well as the raw code outputs, are provided in Supplementary Methods.

References

[1] N. Xi and J. Li. Benchmarking computational doublet-detection methods for single-cell rna sequencing data. *Cell systems*, 2020.

[2] A. Bais and Dennis Kostka. scds: computational annotation of doublets in single-cell rna sequencing data. *Bioinformatics*, 36:1150 – 1158, 2019.

[3] S. Choudhary and R. Satija. Comparison and evaluation of statistical error models for scrna-seq. *Genome Biology*, 23, 2021.

[4] Erica A. K. DePasquale, Erica A. K. DePasquale, Daniel J. Schnell, Pieter-Jan Van Camp, Pieter-Jan Van Camp, igo Valiente-Aland, B. Blaxall, B. Blaxall, H. Grimes, H. Grimes, Harinder Singh, N. Salomonis, and N. Salomonis. Doubletdecon: Deconvoluting doublets from single-cell rna-sequencing data. *Cell reports*, 29:1718 – 1727.e8, 2019.

[5] I. Shainer and M. Stemmer. Choice of alignment pipeline strongly influences clustering quality of scrna-seq datasets. 2021.

[6] Christopher S. McGinnis, Lyndsay M. Murrow, and Zev J. Gartner. Doubletfinder: Doublet detection in single-cell rna sequencing data using artificial nearest neighbors. *bioRxiv*, 2018.

[7] N. Xi and J. Li. Protocol for executing and benchmarking eight computational doublet-detection methods in single-cell rna sequencing data analysis. *STAR Protocols*, 2, 2021.

[8] Pierre-Luc Germain, A. Lun, W. Macnair, and M. Robinson. Doublet identification in single-cell sequencing data using scdblfinder. *F1000Research*, 10, 2021.

[9] Saket Jain, Jonathan Rick, Rushikesh S. Joshi, Angad S. Beniwal, J. Spatz, Sabraj A. Gill, A. Chang, Nikita Choudhary, Alan T. Nguyen, Sweta Sudhir, E. Chalif, Jia-Shu Chen, Ankush Chandra, Alexander F. Haddad, Harsh Wadhwa, Sumedh S. Shah, Serah Choi, J. Hayes, Lin Wang, Garima Yagnik, J. Costello, A. Diaz, D. Heiland, and M. Aghi. Single-cell rna sequencing and spatial transcriptomics reveal cancer-associated fibroblasts in glioblastoma with protumoral effects. *The Journal of Clinical Investigation*, 133, 2023.

[10] S. Freytag, L. Tian, Ingrid Lnnstedt, Milica Ng, and M. Bahlo. Comparison of clustering tools in r for medium-sized 10x genomics single-cell rna-sequencing data. *F1000Research*, 7, 2018.

[11] P. Reyfman, J. Walter, N. Joshi, K. R. Anekalla, Alexandra McQuattie-Pimentel, Stephen Chiu, Ramiro Fernandez, Mahzad Akbarpour, ChingI Chen, Z. Ren, R. Verma, H. Abdala-Valencia, Kiwon Nam, Monica Chi, SeungHye Han, Francisco J. Gonzalez-Gonzalez, S. Soberanes, Satoshi Watanabe, Kinola J. N. Williams, A. S. Flozak, T. Nicholson, Vince K. Morgan, D. Winter, M. Hinchcliff, C. Hrusch, R. Guzy, C. Bonham, A. Sperling, R. Bag, R. Hamanaka, G. Mutlu, A. Yeldandi,

Stacy A. Marshall, A. Shilatifard, L. Amaral, H. Perlman, J. Sznajder, A. Argento, C. Gillespie, J. Dematte, M. Jain, Benjamin D. Singer, K. Ridge, A. Lam, A. Bharat, S. Bhorade, C. Gottardi, G. S. Budinger, and A. Misharin. Single-cell transcriptomic analysis of human lung provides insights into the pathobiology of pulmonary fibrosis. *American Journal of Respiratory and Critical Care Medicine*, 199:1517 – 1536, 2019.

[12] Vitalii Kleshchevnikov, Artem Shmatko, Emma Dann, Alexander Aivazidis, Hamish W. King, Tong Li, A. Lomakin, Veronika R. Kedlian, M. S. Jain, Jun Sung Park, Lauma Ramona, E. Tuck, A. Arutyunyan, R. Vento-Tormo, M. Gerstung, L. James, O. Stegle, and O. Bayraktar. Comprehensive mapping of tissue cell architecture via integrated single cell and spatial transcriptomics. *bioRxiv*, 2020.

[13] Q. Deng, Guangchun Han, N. Puebla-Osorio, M. Ma, P. Strati, B. Chasen, E. Dai, M. Dang, N. Jain, Haopeng Yang, Yuanxin Wang, Shaojun Zhang, Ruiping Wang, Runzhe Chen, Jordan Showell, Sreejoyee Ghosh, Sridevi Patchva, Qi Zhang, R. Sun, F. Hagemeister, L. Fayad, F. Samaniego, Hans C. Lee, L. Nastoupil, N. Fowler, R. Eric Davis, J. Westin, S. Neelapu, Linghua Wang, and Michael R. Green. Characteristics of anti-cd19 car t cell infusion products associated with efficacy and toxicity in patients with large b cell lymphomas. *Nature Medicine*, 26:1878 – 1887, 2020.

[14] Tianhui Shi, Mingshu Zhai, Yi Xu, and Jidong Zhai. Graphpi: High performance graph pattern matching through effective redundancy elimination. *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14, 2020.

[15] Samuel L. Wolock, Romain Lopez, and Allon M. Klein. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *bioRxiv*, 2018.

[16] Qian Zhu, Sheel Shah, Ruben Dries, L. Cai, and Guocheng Yuan. Identification of spatially associated subpopulations by combining scrna-seq and sequential fluorescence in situ hybridization data. *Nature biotechnology*, 2018.

A Data Description

Here is the data description, as provided by the user:

```
\#\# General Description
The dataset includes results from various doublet detection
    algorithms applied to multiple scRNA-seq datasets with
    different doublet content and algorithm parameters. While
    scRNA-seq aims to measure the transcriptomes of individual
    cells, doublets can occur when two cells are captured as
    one. The purpose of doublet detection algorithms is to
    accurately identify these doublets so they can be removed
    for downstream analysis. The four algorithms evaluated are
    DoubletFinder, hybrid, scDblFinder, and Scrublet. Key
    performance metrics for these algorithms include the area
    under the precision-recall curve (AUPRC), the area under
    the receiver operating characteristic curve (AUROC), and
    the true negative rate (TNR).
\#\# Data Files
The dataset consists of 2 data files:

\#\#\# File 1: "barcodedNonBarcoded\_AUPRC\_AUROC\_TNR.csv"
The CSV file contains a dataset with each row representing a
    result and each column representing a feature. The columns
    in the dataset are as follows:

"": Index number
"X": Index number
"dataset": The scRNA-seq dataset from which the data originated
"sample": The specific sample within the corresponding scRNA-
    seq dataset from which the data originated
"condition": The doublet detection algorithm used to generate
    the AUPRC, AUROC, and TNR data (options include
    DoubletFinder, hybrid, scDblFinder, or Scrublet)
"auprc": Area under the precision-recall curve
"auroc": Area under the receiver operating characteristic curve
"dbl\_act": Actual true doublet rate of the dataset
"isBarcoded": Indicates whether barcoding technology was used
    in the dataset
"TNR": True negative rate


Here are the first few lines of the file:
'''output
"","X","dataset","sample","condition","auprc","auroc","dbl\_act
    ","isBarcoded","TNR"
"1",1,"Jain et al.","1\_DMSO\_A","DoubletFinder
    ",0.178563957295622,0.706881063600431,0.08,"Barcoded
    ",0.931927975406236
```

```
"2",2,"Jain et al.","2\_DMSO\_B","DoubletFinder
    ",0.202027089060359,0.721426428678566,0.08,"Barcoded
    ",0.933136676499508
"3",3,"Jain et al.","3\_LSD1i\_A","DoubletFinder
    ",0.187261326956338,0.660830700566672,0.08,"Barcoded
    ",0.927109974424553

```

```

\#\#\# File 2: "TNR\_plotted\_formatted.csv"
The CSV file contains a dataset with each row representing a
    result and each column representing a feature. The columns
    in the dataset are as follows:

"": Index number
"X": Index number
"condition": The doublet detection algorithm used to generate
    the AUPRC, AUROC, and TNR data (options include
    DoubletFinder, hybrid, scDblFinder, or Scrublet)
"dataset": The scRNA-seq dataset from which the data originated
"sample": The specific sample within the corresponding scRNA-
    seq dataset
"TNR": True negative rate
"dbl\_exp": Expected true doublet rate from 10X Genomics
"dbl\_act": Actual true doublet rate from 10X Genomics


Here are the first few lines of the file:
```output
"","X","condition","dataset","sample","TNR","dbl_exp","dbl\
    _act"
"1",1,"DoubletFinder","Jain et al.","1_DMSO_A
    ",0.931488801054018,0.05,0.08
"2",2,"DoubletFinder","Jain et al.","1_DMSO_A
    ",0.936758893280632,0.08,0.08
"3",3,"DoubletFinder","Jain et al.","1_DMSO_A
    ",0.928853754940711,0.1,0.08

```
```

B Data Exploration

B.1 Code

The Data Exploration was carried out using the following custom code:

```
import pandas as pd

def summarize_data(file_paths, output_file):
    summary = []

    for file_path in file_paths:
        # Load the dataset
        data = pd.read_csv(file_path)

        summary.append(f"# Summary of {file_path}\n")

        # Data Size
        summary.append("# Data Size\n")
        summary.append(f"Number of Rows: {data.shape[0]}\n")
        summary.append(f"Number of Columns: {data.shape[1]}\n")
        summary.append("\n")

        # Summary Statistics
        summary.append("# Summary Statistics\n")
        summary.append(data.describe().to_string())
        summary.append("\n\n")

        # Categorical Variables
        summary.append("# Categorical Variables\n")
        categorical_cols = data.select_dtypes(include=['object'
            ↪ ]).columns
        if len(categorical_cols) > 0:
            for col in categorical_cols:
                summary.append(f"Column: {col}\n")
                most_common_val = data[col].mode()[0]
                summary.append(f"Most Common Value: {
                    ↪ most_common_val}\n")
                summary.append("\n")
        else:
            summary.append("# Not Applicable\n")
        summary.append("\n")

        # Missing Values
        summary.append("# Missing Values\n")
        missing_values = data.isnull().sum()
        for col, count in missing_values.items():
            summary.append(f"{col}: {count}\n")

        # Check for any special numeric values that stand for
            ↪ unknown/undefined
        special_numeric = data.isin([-1, 9999]).sum()
        if special_numeric.any():
            summary.append("\n# Special Numeric Values
                ↪ Indicating Unknown/Undefined\n")
```

```
                for col, count in special_numeric.items():
                    if count > 0:
                        summary.append(f"{col}: {count}\n")
            summary.append("\n")

    # Write the summary to the output file
    with open(output_file, "w") as f:
        f.writelines(summary)

# File paths
files = ["barcodedNonBarcoded_AUPRC_AUROC_TNR.csv", "
    ↪ TNR_plotted_formatted.csv"]
output_file = "data_exploration.txt"

# Summarize data for both files
summarize_data(files, output_file)
```

B.2 Code Description

The provided code conducts a comprehensive exploration of two CSV datasets comprising results from various doublet detection algorithms applied to single-cell RNA sequencing (scRNA-seq). Doublet detection is crucial in scRNA-seq analysis because doublets—instances where two cells are captured together—can distort the interpretation of cellular transcriptomic data. The main function, `summarize_data`, processes each dataset and generates a structured summary, facilitating downstream analyses.

The code begins by evaluating each dataset's size, reporting the total number of rows and columns. This information is essential for understanding the overall scale of the datasets and their complexity. Next, the function generates summary statistics for numeric columns, including count, mean, standard deviation, minimum, and maximum values. These metrics not only provide insights into the distributions and ranges of various performance measures, such as the Area Under the Precision-Recall Curve (AUPRC) and Area Under the Receiver Operating Characteristic Curve (AUROC), but also help in identifying any anomalies and assessing the quality of the data.

The code then identifies and summarizes categorical variables, calculating the most common value (mode) for each. This analysis sheds light on the dominant experimental conditions or algorithmic performances across the different scRNA-seq datasets.

Additionally, the presence of missing values in each dataset is assessed, with a count of NaN (not a number) entries recorded for each column. Un-

derstanding the extent of missing data is critical, as it can lead to potential biases and a reduction in statistical power, which may influence the validity of conclusions drawn from the analyses. The code also checks for any special numeric values—commonly used to signify unknown or undefined data—and quantifies their occurrences.

The summary outputs, encompassing dataset size, summary statistics, categorical variable insights, missing values, and any special numeric values, are compiled and written to a text file named `data_exploration.txt`. The structured output includes clearly demarcated sections with headings, which facilitate easy navigation through the generated report. For instance, sections named "# Data Size" report the number of rows and columns, while "# Summary Statistics" details the computed metrics. Utilizing a consistent naming convention for output files, such as "data_exploration.txt," aids in organizing results for multiple analyses, particularly in larger projects. This organized structure ultimately enhances the utility of the summary for researchers undertaking further experimentation or interpretation efforts.

B.3 Code Output

data_exploration.txt

```
\# Summary of barcodedNonBarcoded\_AUPRC\_AUROC\_TNR.csv
\# Data Size
Number of Rows: 396
Number of Columns: 10

\# Summary Statistics
       Unnamed: 0     X   auprc  auroc  dbl\_act    TNR
count         396   396     396    396       348    396
mean        198.5 198.5  0.2987 0.7499      0.08 0.9419
std         114.5 114.5  0.1562 0.1556 1.39e-17 0.0128
min             1     1 0.05269 0.2352      0.08 0.9154
25\%         99.75 99.75  0.1836 0.7042      0.08 0.9334
50\%         198.5 198.5  0.3011 0.8044      0.08 0.9422
75\%         297.2 297.2  0.3838 0.8495      0.08 0.9501
max            396   396  0.9705 0.9961      0.08 0.9923

\# Categorical Variables
Column: dataset
Most Common Value: LARRY

Column: sample
Most Common Value: 1-1uMPLX

Column: condition
```

```
Most Common Value: DoubletFinder

Column: isBarcoded
Most Common Value: Barcoded


\# Missing Values
Unnamed: 0: 0
X: 0
dataset: 0
sample: 48
condition: 0
auprc: 0
auroc: 0
dbl\_act: 48
isBarcoded: 0
TNR: 0

\# Summary of TNR\_plotted\_formatted.csv
\# Data Size
Number of Rows: 2088
Number of Columns: 8

\# Summary Statistics
       Unnamed: 0     X      TNR  dbl\_exp   dbl\_act
count        2088  2088     2088      2088       2088
mean         1044  1150   0.9409    0.1383       0.08
std         602.9 672.5  0.01202   0.06964  1.388e-17
min             1     1   0.9148      0.05       0.08
25\%        522.8 582.8   0.9333      0.08       0.08
50\%         1044  1164   0.9415     0.125       0.08
75\%         1566  1746   0.9494       0.2       0.08
max          2088  2328   0.9772      0.25       0.08

\# Categorical Variables
Column: condition
Most Common Value: DoubletFinder

Column: dataset
Most Common Value: LARRY

Column: sample
Most Common Value: 1-1uMPLX


\# Missing Values
Unnamed: 0: 0
X: 0
condition: 0
```

```
dataset: 0
sample: 0
TNR: 0
dbl\_exp: 0
dbl\_act: 0
```

C Data Analysis

C.1 Code

The Data Analysis was carried out using the following custom code:

```
# IMPORT
import pandas as pd
import numpy as np
import pickle
from scipy.stats import f_oneway, kruskal
from sklearn.preprocessing import LabelEncoder

# LOAD DATA
file1 = pd.read_csv('barcodedNonBarcoded_AUPRC_AUROC_TNR.csv')
file2 = pd.read_csv('TNR_plotted_formatted.csv')

# DATASET PREPARATIONS
df = pd.merge(file1, file2, on=['sample', 'dataset', 'condition
    ↪ '], how='inner')
df = df.dropna()

# DESCRIPTIVE STATISTICS
## Table 0: "Descriptive statistics of AUPRC, AUROC, and TNR
    ↪ stratified by condition"
desc_stat = df.groupby('condition')[['auprc', 'auroc', 'TNR_x',
    ↪  'dbl_act_x', 'dbl_exp']].mean()
desc_stat.columns = ['Mean AUPRC', 'Mean AUROC', 'Mean TNR', '
    ↪ Mean dbl_act', 'Mean dbl_exp']
desc_stat_std = df.groupby('condition')[['auprc', 'auroc', '
    ↪ TNR_x', 'dbl_act_x', 'dbl_exp']].std()
desc_stat_std.columns = ['STD AUPRC', 'STD AUROC', 'STD TNR', '
    ↪ STD dbl_act', 'STD dbl_exp']
df0 = pd.concat([desc_stat, desc_stat_std], axis=1)
df0.to_pickle('table_0.pkl')

# PREPROCESSING
labelencoder = LabelEncoder()
df['condition_code'] = labelencoder.fit_transform(df['condition
    ↪ '])

# ANALYSIS
```

```
## Table 1: "ANOVA results comparing AUPRC across algorithms"
auprc_results = f_oneway(df['auprc'][df['condition'] == '
    ↪ DoubletFinder'],
                         df['auprc'][df['condition'] == 'hybrid
                             ↪ '],
                         df['auprc'][df['condition'] == '
                             ↪ scDblFinder'],
                         df['auprc'][df['condition'] == '
                             ↪ Scrublet'])
df1 = pd.DataFrame({'F-value': [auprc_results.statistic], 'P-
    ↪ value': [auprc_results.pvalue]}, index=['AUPRC'])
df1['Best Algorithm'] = desc_stat['Mean AUPRC'].idxmax()
df1.to_pickle('table_1.pkl')

## Table 2: "ANOVA results comparing AUROC across algorithms"
auroc_results = f_oneway(df['auroc'][df['condition'] == '
    ↪ DoubletFinder'],
                         df['auroc'][df['condition'] == 'hybrid
                             ↪ '],
                         df['auroc'][df['condition'] == '
                             ↪ scDblFinder'],
                         df['auroc'][df['condition'] == '
                             ↪ Scrublet'])
df2 = pd.DataFrame({'F-value': [auroc_results.statistic], 'P-
    ↪ value': [auroc_results.pvalue]}, index=['AUROC'])
df2['Best Algorithm'] = desc_stat['Mean AUROC'].idxmax()
df2.to_pickle('table_2.pkl')

## Table 3: "Kruskal-Wallis results comparing TNR across
    ↪ algorithms"
tnr_results = kruskal(df['TNR_x'][df['condition'] == '
    ↪ DoubletFinder'],
                      df['TNR_x'][df['condition'] == 'hybrid'],
                      df['TNR_x'][df['condition'] == '
                          ↪ scDblFinder'],
                      df['TNR_x'][df['condition'] == 'Scrublet'
                          ↪ ])
df3 = pd.DataFrame({'H-value': [tnr_results.statistic], 'P-
    ↪ value': [tnr_results.pvalue]}, index=['TNR'])
df3['Best Algorithm'] = desc_stat['Mean TNR'].idxmax()
df3.to_pickle('table_3.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': df.shape[0]
}

with open('additional_results.pkl', 'wb') as f:
```

```
pickle.dump(additional_results, f)
```

C.2 Code Description

The provided code performs a comprehensive analysis of doublet detection performance in single-cell RNA sequencing (scRNA-seq) datasets using four algorithms: DoubletFinder, hybrid, scDblFinder, and Scrublet. The analysis process can be summarized as follows:

Data Integration and Preprocessing The two provided CSV files—`barcodedNonBarcoded_AUPRC_` and `TNR_plotted_formatted.csv`—are read into Pandas dataframes and subsequently merged based on common columns including `sample`, `dataset`, and `condition`. This integration step is crucial for obtaining a comprehensive dataset for downstream analysis. Following the merge, rows containing missing values are dropped, and categorical labels for the `condition` (algorithm) column are encoded into numeric format for ease of statistical computations.

Descriptive Statistics Descriptive statistics are calculated to summarize the performance of the doublet detection algorithms. These statistics include the mean and standard deviation of the Area Under the Precision-Recall Curve (AUPRC), Area Under the Receiver Operating Characteristic Curve (AUROC), True Negative Rate (TNR), as well as the actual and expected true doublet rates. The results are grouped by the `condition` column (algorithm) and stored in a Pandas dataframe which is then saved into a file `table_0.pkl`.

Statistical Analysis Three distinct statistical tests are conducted to compare the performance of the algorithms:

AUPRC Analysis An Analysis of Variance (ANOVA) test is performed to compare the AUPRC across the four algorithms. This test assesses whether there are any statistically significant differences in the mean AUPRC values.

AUROC Analysis Similarly, an ANOVA is conducted to compare the AUROC values across the algorithms. This test evaluates if differences in the mean AUROC values are statistically significant.

TNR Analysis A Kruskal-Wallis H test, a non-parametric equivalent of ANOVA, is applied to compare the TNR across the algorithms. This test is particularly useful for datasets that do not meet the assumptions required for ANOVA.

For each of these tests, the F-value (for ANOVAs) or H-value (for Kruskal-Wallis) and the corresponding p-value are computed and stored in separate dataframes. Additionally, the algorithm with the highest mean value for each metric (AUPRC, AUROC, TNR) is identified and recorded. These results are saved into files `table_1.pkl`, `table_2.pkl`, and `table_3.pkl` respectively.

Additional Information An additional file `additional_results.pkl` is generated, which contains the total number of observations in the processed dataset. This provides context for the statistical power of the analyses performed.

C.3 Code Output

table_0.pkl

```
                Mean AUPRC  Mean AUROC  Mean TNR  Mean dbl\_act
                     Mean dbl\_exp  STD AUPRC  STD AUROC  STD
                    TNR  STD dbl\_act  STD dbl\_exp
condition
DoubletFinder       0.3368      0.8066     0.944          0.08
           0.1383     0.1079    0.05711 0.008325            0
         0.06969
Scrublet            0.3439      0.8145     0.949          0.08
           0.1383     0.1201    0.06042 0.008747            0
         0.06969
hybrid              0.3399      0.8492     0.943          0.08
           0.1383    0.09282    0.05157 0.007075            0
         0.06969
scDblFinder         0.1273      0.5262    0.9278          0.08
           0.1383     0.1072     0.1542  0.01029            0
         0.06969
```

table_1.pkl

```
       F-value    P-value Best Algorithm
AUPRC    513.3  1.03e-249       Scrublet
```

table_2.pkl

```
       F-value P-value Best Algorithm
AUROC     1404       0         hybrid
```

table_3.pkl

```
     H-value    P-value Best Algorithm
TNR      929  4.64e-201       Scrublet
```

additional_results.pkl

```
{
    'Total number of observations': 2088,
}
```

D LaTeX Table Design

D.1 Code

The LaTeX Table Design was carried out using the following custom code:

```
# IMPORT
import pandas as pd
from my_utils import to_latex_with_note, is_str_in_df,
    ↪ split_mapping, AbbrToNameDef
from typing import Optional, Dict, Any, Tuple

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {
    'Mean AUPRC': ('Avg. AUPRC', 'Average Area Under Precision-
        ↪ Recall Curve'),
    'Mean AUROC': ('Avg. AUROC', 'Average Area Under Receiver
        ↪ Operating Characteristics Curve'),
    'Mean TNR': ('Avg. TNR', 'Average True Negative rate'),
    'STD AUPRC': ('StdDev AUPRC', 'Standard Deviation of AUPRC'
        ↪ ),
    'STD AUROC': ('StdDev AUROC', 'Standard Deviation of AUROC'
        ↪ ),
    'STD TNR': ('StdDev TNR', 'Standard Deviation of TNR'),
    'Mean dbl_act': ('Avg. Act. Doublet Rate', 'Average actual
        ↪ doublet rate'),
    'Mean dbl_exp': ('Avg. Exp. Doublet Rate', 'Average
        ↪ expected doublet rate'),
    'STD dbl_act': ('StdDev Act. Doublet Rate', 'Standard
        ↪ Deviation of the actual doublet rate'),
    'STD dbl_exp': ('StdDev Exp. Doublet Rate', 'Standard
        ↪ Deviation of the expected doublet rate'),
    'condition': (None, 'Condition Applied: DoubletFinder,
        ↪ hybrid, scDblFinder or Scrublet')
```

```
}

# TABLE 0:
df0 = pd.read_pickle('table_0.pkl')

# TRANSPOSE THE DATASET
df0 = df0.T

# RENAME ROWS AND COLUMNS
mapping0 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df0, k))
mapping0 |= {
    'condition': ('Condition', None),
    'DoubletFinder': ('DbltFndr', 'Doublet Finder Algorithm'),
    'Scrublet': ('Scrblt', 'Scrublet Algorithm'),
    'hybrid': ('hyb', 'Hybrid Algorithm'),
    'scDblFinder': ('scDBlFndr', 'scDblFinder Algorithm'),
}
abbrs_to_names0, legend0 = split_mapping(mapping0)
df0 = df0.rename(columns=abbrs_to_names0, index=abbrs_to_names0
    ↪ )

# SAVE AS LATEX:
to_latex_with_note(
    df0, 'table_0.tex',
    caption='Descriptive statistics of performance metrics
        ↪ stratified by condition',
    label='table:table_0',
    note=None,
    legend=legend0)

# TABLE 1:
df1 = pd.read_pickle('table_1.pkl')

# RENAME ROWS AND COLUMNS
mapping1 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df1, k))
mapping1 |= {'AUPRC': ('AUPRC', 'Area Under Precision-Recall
    ↪ Curve')}

abbrs_to_names1, legend1 = split_mapping(mapping1)
df1 = df1.rename(columns=abbrs_to_names1, index=abbrs_to_names1
    ↪ )

# SAVE AS LATEX:
to_latex_with_note(
    df1, 'table_1.tex',
    caption='ANOVA results comparing Area Under Precision-
        ↪ Recall Curve (AUPRC) across algorithms',
```

```
    label='table:table_1',
    note=None,
    legend=legend1)

# TABLE 2:
df2 = pd.read_pickle('table_2.pkl')

# RENAME ROWS AND COLUMNS
mapping2 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df2, k))
mapping2 |= {'AUROC': ('AUROC', 'Area Under Receiver Operating
    ↪ Characteristics Curve')}

abbrs_to_names2, legend2 = split_mapping(mapping2)
df2 = df2.rename(columns=abbrs_to_names2, index=abbrs_to_names2
    ↪ )

# SAVE AS LATEX:
to_latex_with_note(
    df2, 'table_2.tex',
    caption='ANOVA results comparing Area Under Receiver
        ↪ Operating Characteristic (AUROC) across algorithms',
    label='table:table_2',
    note=None,
    legend=legend2)

# TABLE 3:
df3 = pd.read_pickle('table_3.pkl')

# RENAME ROWS AND COLUMNS
mapping3 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df3, k))
mapping3 |= {'TNR': ('TNR', 'True Negative Rate')}

abbrs_to_names3, legend3 = split_mapping(mapping3)
df3 = df3.rename(columns=abbrs_to_names3, index=abbrs_to_names3
    ↪ )

# SAVE AS LATEX:
to_latex_with_note(
    df3, 'table_3.tex',
    caption='Kruskal-Wallis results comparing True Negative
        ↪ Rate (TNR) across algorithms',
    label='table:table_3',
    note=None,
    legend=legend3)
```

D.2 Provided Code

The code above is using the following provided functions:

```
def to_latex_with_note(df, filename: str, caption: str, label:
    ↪ str, note: str = None, legend: Dict[str, str] = None, **
    ↪ kwargs):
    """
    Converts a DataFrame to a LaTeX table with optional note
        ↪ and legend added below the table.

    Parameters:
    - df, filename, caption, label: as in `df.to_latex`.
    - note (optional): Additional note below the table.
    - legend (optional): Dictionary mapping abbreviations to
        ↪ full names.
    - **kwargs: Additional arguments for `df.to_latex`.
    """

def is_str_in_df(df: pd.DataFrame, s: str):
    return any(s in level for level in getattr(df.index, '
        ↪ levels', [df.index]) + getattr(df.columns, 'levels',
        ↪ [df.columns]))

AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]

def split_mapping(abbrs_to_names_and_definitions: AbbrToNameDef
    ↪ ):
    abbrs_to_names = {abbr: name for abbr, (name, definition)
        ↪ in abbrs_to_names_and_definitions.items() if name is
        ↪ not None}
    names_to_definitions = {name or abbr: definition for abbr,
        ↪ (name, definition) in abbrs_to_names_and_definitions.
        ↪ items() if definition is not None}
    return abbrs_to_names, names_to_definitions
```

D.3 Code Output

table_0.tex

```
\% This latex table was generated from: `table\_0.pkl`
\begin{table}[h]
\caption{Descriptive statistics of performance metrics
    stratified by condition}
\label{table:table\_0}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{\%
```

```
\begin{tabular}{lrrrr}
\toprule
condition \& DbltFndr \& Scrblt \& hyb \& scDBlFndr \\
\midrule
\textbf{Avg. AUPRC} \& 0.337 \& 0.344 \& 0.34 \& 0.127 \\
\textbf{Avg. AUROC} \& 0.807 \& 0.815 \& 0.849 \& 0.526 \\
\textbf{Avg. TNR} \& 0.944 \& 0.949 \& 0.943 \& 0.928 \\
\textbf{Avg. Act. Doublet Rate} \& 0.08 \& 0.08 \& 0.08 \& 0.08
    \\
\textbf{Avg. Exp. Doublet Rate} \& 0.138 \& 0.138 \& 0.138 \&
   0.138 \\
\textbf{StdDev AUPRC} \& 0.108 \& 0.12 \& 0.0928 \& 0.107 \\
\textbf{StdDev AUROC} \& 0.0571 \& 0.0604 \& 0.0516 \& 0.154 \\
\textbf{StdDev TNR} \& 0.00832 \& 0.00875 \& 0.00707 \& 0.0103
   \\
\textbf{StdDev Act. Doublet Rate} \& 0 \& 0 \& 0 \& 0 \\
\textbf{StdDev Exp. Doublet Rate} \& 0.0697 \& 0.0697 \& 0.0697
    \& 0.0697 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Avg. AUPRC}: Average Area Under Precision-Recall
   Curve
\item \textbf{Avg. AUROC}: Average Area Under Receiver
   Operating Characteristics Curve
\item \textbf{Avg. TNR}: Average True Negative rate
\item \textbf{StdDev AUPRC}: Standard Deviation of AUPRC
\item \textbf{StdDev AUROC}: Standard Deviation of AUROC
\item \textbf{StdDev TNR}: Standard Deviation of TNR
\item \textbf{Avg. Act. Doublet Rate}: Average actual doublet
   rate
\item \textbf{Avg. Exp. Doublet Rate}: Average expected doublet
    rate
\item \textbf{StdDev Act. Doublet Rate}: Standard Deviation of
   the actual doublet rate
\item \textbf{StdDev Exp. Doublet Rate}: Standard Deviation of
   the expected doublet rate
\item \textbf{DbltFndr}: Doublet Finder Algorithm
\item \textbf{Scrblt}: Scrublet Algorithm
\item \textbf{hyb}: Hybrid Algorithm
\item \textbf{scDBlFndr}: scDblFinder Algorithm
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\% This latex table was generated from: `table\_1.pkl`
```

```
\begin{table}[h]
\caption{ANOVA results comparing Area Under Precision-Recall
    Curve (AUPRC) across algorithms}
\label{table:table\_1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{\%
\begin{tabular}{lrll}
\toprule
 \& F-value \& P-value \& Best Algorithm \\
\midrule
\textbf{AUPRC} \& 513 \& \$$<$\$1e-06 \& Scrublet \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{AUPRC}: Area Under Precision-Recall Curve
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\% This latex table was generated from: `table\_2.pkl`
\begin{table}[h]
\caption{ANOVA results comparing Area Under Receiver Operating
    Characteristic (AUROC) across algorithms}
\label{table:table\_2}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{\%
\begin{tabular}{lrll}
\toprule
 \& F-value \& P-value \& Best Algorithm \\
\midrule
\textbf{AUROC} \& 1.4e+03 \& \$$<$\$1e-06 \& hybrid \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{AUROC}: Area Under Receiver Operating
    Characteristics Curve
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_3.tex

```
\% This latex table was generated from: `table\_3.pkl`
```

```
\begin{table}[h]
\caption{Kruskal-Wallis results comparing True Negative Rate (
    TNR) across algorithms}
\label{table:table\_3}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{\%
\begin{tabular}{lrll}
\toprule
 \& H-value \& P-value \& Best Algorithm \\
\midrule
\textbf{TNR} \& 929 \& \$$<$\$1e-06 \& Scrublet \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{TNR}: True Negative Rate
\end{tablenotes}
\end{threeparttable}
\end{table}
```