Silesian University of Technology
Faculty of Automatic Control, Electronics
and Computer Science

Skipping batch effect correction: clustering-based methods for analyzing confounded single-cell RNA-sequencing data

PhD Thesis

Author:	**Tomasz Kujawa**
Supervisor:	**prof. dr hab. inż. Joanna Polańska**
Co-supervisor:	**dr inż. Michał Marczyk**

Gliwice, June 2023

Contents:

I. INTRODUCTION

I.1 Motivation

Analysis of data from single-cell RNA sequencing experiments (scRNAseq) is a challenging task due to several reasons. The major problems involve high dimensionality of the data and measurement noise of various origins. A typical scRNAseq dataset consists of tens of thousands dimensions (the dimension corresponds to the number of genes) measured across hundreds of thousands of cells. Each dimension carries a substantial level of technical noise resulting from variations introduced during data generation, and biological noise resulting from natural differences between cells and cell types. As a result, scRNA-seq data exhibits a high fraction of zero measurements, often referred to as sparsity.

ScRNAseq experiments are often conducted on a large scale, involving multiple laboratories or measurements taken at different times. Perfectly balanced experimental designs for such large projects may be infeasible, resulting in the need to conduct experiments in batches. Consequently, batch effects inevitably arise. Batch effects introduce variation that is unrelated to the biological variability under investigation, thereby obscuring it. If left unaddressed, batch effects can result in misleading conclusions drawn from the analysis. Although batch effects can be easily detected in high-dimensional data, confounding factors leading to them may not be recorded during the course of an experiment. Therefore, batch effects have to be computationally corrected or removed.

Numerous approaches based on different ideas and assumptions have been proposed in the literature, but gold standards have yet to be established. The main challenge lies in distinguishing biological from technical variability which often results in overcorrection, meaning the removal of biological differences between batches. This is mainly because researchers often lack prior knowledge of the underlying cell types before conducting an experiment. Furthermore, downstream gene-level analyses are not safe to be performed on corrected data because in most cases correction distorts the original data distribution, and there is lack of a measure to quantify the uncertainty associated with the correction process. Therefore, there is a strong need to develop research in the field of batch effect removal, correction or mitigation employing new approaches and bioinformatic tools. This task is currently of high priority and considered one of the grand challenges in scRNAseq data analysis. It serves as the primary motivation for this thesis.

I.2 Aim and theses of this work

This work aims to provide a pipeline that utilizes iterative subspace clustering, combined with functional analysis of gene sets, to mitigate the negative impact of the batch effect on scRNAseq data. The crucial aspect of the functional analysis involves identifying cluster-specific pathways and establishing their linkage between batches. Therefore, the proposed workflow eliminates the need for applying batch-effect correction and enables consolidated analysis of batches that were generated separately. In contrast to existing complex and computationally demanding algorithms, this approach prioritizes simplicity, low computational cost, and ease of interpretation. The utilization of subspace clustering combined with functional analysis of gene pathways for mitigating batch effects has not been explored before, making this thesis a novel contribution to the field.

The underlying assumption is that iterative subspace clustering may diminish batch effects by removing more noise from the data with each subsequent iteration. As a result, cells should tend to form groups based on their true biology. Furthermore, the cluster-specific pathways identified are expected to exhibit robust manifestations and demonstrate resilience to the negative impact of batch effects, which is typically less pronounced compared to individual genes.

The theses of this dissertation are formulated as follows:

1. Existing algorithms for batch effect correction in scRNAseq often distort the original distribution of gene expression data. Consequently, gene-level analyses such as differential expression or marker identification cannot be safely applied to the corrected dataset.

2. A simple feature selection strategy based on variance decomposition yields similar results to more sophisticated and computationally expensive methods.

3. In confounded scRNA-seq data, batch effect correction can be skipped. Instead, a reliable analysis can be performed by independently identifying subclusters of cells within each batch and then linking them between batches based on the similarity of their functional profiles to track similar cells from different batches.

I.3 Thesis structure

The thesis is structured in the following way:

- **Chapter I** provides an overview of how a single-cell RNA-seq experiment is conducted, with an emphasis on the sources of noise in scRNA-seq data. It also discusses current approaches to noise reduction in scRNA-seq data.

- **Chapter II** describes experimental design, datasets, and the methods used for data generation and analysis.

- **Chapter III** presents the results of evaluation of various batch correction algorithms.

- **Chapter IV** showcases the results obtained with the proposed workflow.

- **Chapter V** provides a discussion and summary of this work.

II. BACKGROUND

II.1 scRNAseq - technology and its applications

Molecular analyses of a cell can be conducted using various approaches, encompassing different levels known as omics layers [1]. Each omics layer focuses on measuring distinct biomolecules, such as DNA molecules at the genome level, mRNA sequences at the transcriptome level, or proteins at the proteome level (**Figure 1**) [2]. There exists a continuous process of information transmission among these omics layers: from DNA to RNA, from RNA to protein, and from protein to biological pathways. The culmination of these multi-omics molecular interactions manifests as the cellular phenotype [3].

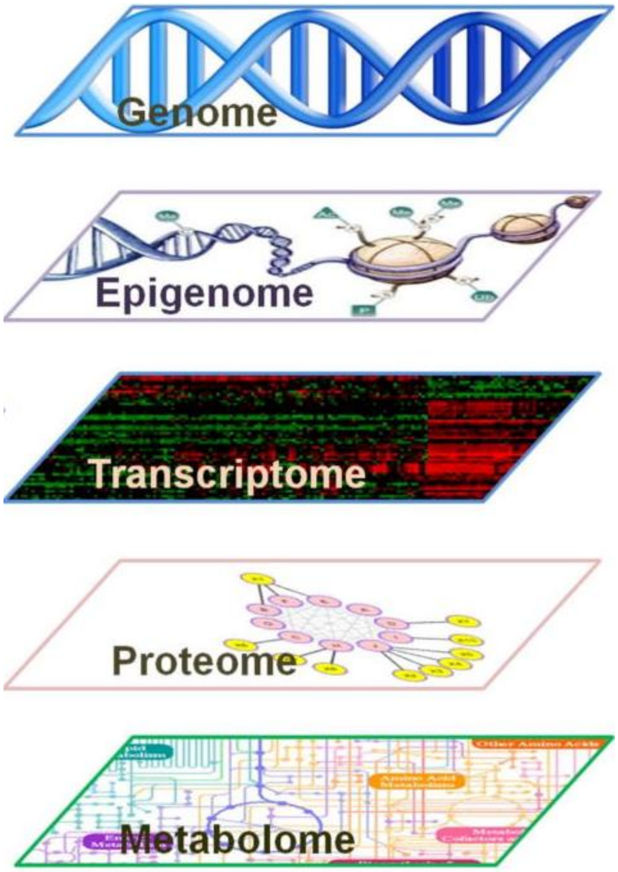


Figure 1. *Molecular layers of cell identity [2].*

While multicellular organisms have the same set of genes (genomes) across most cells, the expression of these genes varies between cells, even among those of the same type. This indicates that different cells exhibit distinct patterns of gene expression [4, 5]. The transcriptional activity of a cell change as it ages or progresses through different developmental stages [6, 7]. Additionally, many genes are expressed in response to various environmental factors [8]. Therefore, the transcriptome serves as a reliable representative of the cellular state, offering a direct read-out of the dynamic decision-making processes within a cell [9-11].

One strategy for quantifying the RNA content in a sample is through RNA sequencing (RNAseq). Initially, transcriptomes were analyzed in populations of cells derived from a specific tissue. This approach, often referred to as 'bulk RNAseq,' was motivated by the assumption that cells from the same tissue type are homogeneous. As a result, the obtained results encompass a mixture of different gene expression profiles from the population of cells under investigation. In other words, the expression signal is averaged across all cells. While the bulk approach is sufficient for characterizing the overall state of a tissue, it completely masks the signal originating from individual cells and overlooks tissue heterogeneity (**Figure 2** – bottom panel).

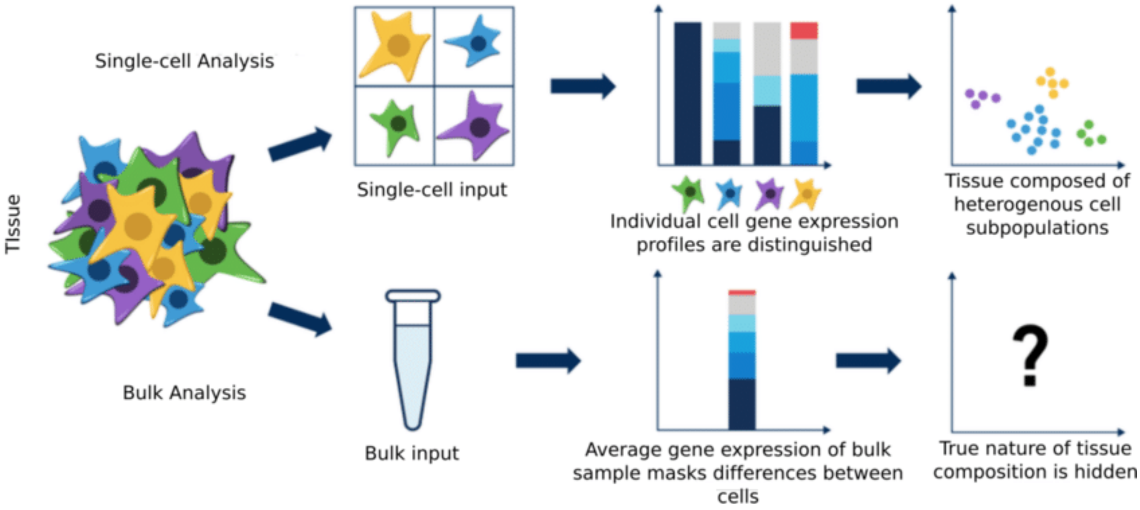

Single-cell Analysis
Tissue
Single-cell input
Individual cell gene expression profiles are distinguished
Tissue composed of heterogenous cell subpopulations
Bulk Analysis
Bulk input
Average gene expression of bulk sample masks differences between cells
?
True nature of tissue composition is hidden

Figure 2. *Difference between scRNAseq and bulk RNAseq [13].*

The barrier of single-cell was breached in 2009 with the emergence of the first publication on single-cell RNA sequencing (scRNAseq) [12]. Since then, it has become possible to study the cell type-specific contribution to the expression profile of a sample (**Figure 2 -** upper panel).

Nowadays, scRNAseq allows for the parallel processing of millions of individual cells simultaneously, enabling the assessment of transcriptional differences between any pair of them [14]. This technique has revolutionized the field of life sciences leading to ground-breaking discoveries in developmental biology [15], immunology [5], neuroscience [16] and oncology [17, 18]. The significant impact of scRNAseq in unraveling cell functions led to its recognition as the "Method of the Year" by the journal Nature Methods in 2013 [19]. In 2019, the same distinction was awarded to the combination of scRNAseq with protein profiling [20].

The investigation of heterogeneity is a fundamental focus of the scRNAseq research [21]. Its primary objective is to identify subpopulations of cells within healthy tissues or cancer cells [22, 23]. One of the ground-breaking discoveries in this field was the identification of a new and rare type of cell in the human airway known as the pulmonary ionocytes [24]. It is believed that these cells are responsible for the development of cystic fibrosis (**Figure 3**).

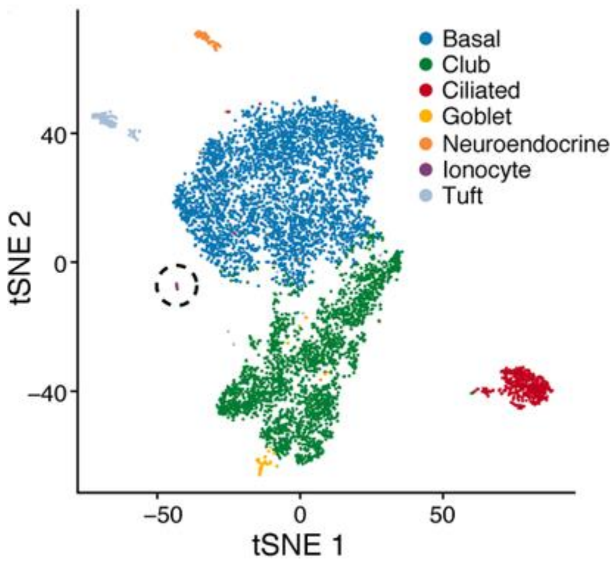


Figure 3. *Discovery of pulmonary ionocytes with scRNAseq [25]. The plot presents t-distributed stochastic neighbor embedding (t-SNE) colored by cell type. The novel ionocyte cluster is circled.*

Heterogeneity studies using scRNAseq technology also focus on the identification of tumour biomarkers [25, 26] as well as therapeutic targets and resistance pathways [27, 28]. The ability to study cellular heterogeneity enabled by scRNAseq technology has led to the development of large-scale projects intending to construct cell atlases for tissues, organs, and even entire organisms. Examples include the Human Cell Atlas (*H. sapiens*) [29], Tabula Muris (*M. musculus*) [30], and Fly Cell Atlas (*D. melanogaster*) [31].

ScRNAseq can provide insights into a common question in biology: how cells transition from one state to another during various biological processes, such as development, differentiation, or in response to external stimuli [10, 18]. This is the focus of trajectory inference or pseudotime analysis, which aims to reconstruct a path (trajectory) that describes the transitions of a cell between different developmental states or its differentiation into increasingly specialized cell subtypes (**Figure 4**) [15].

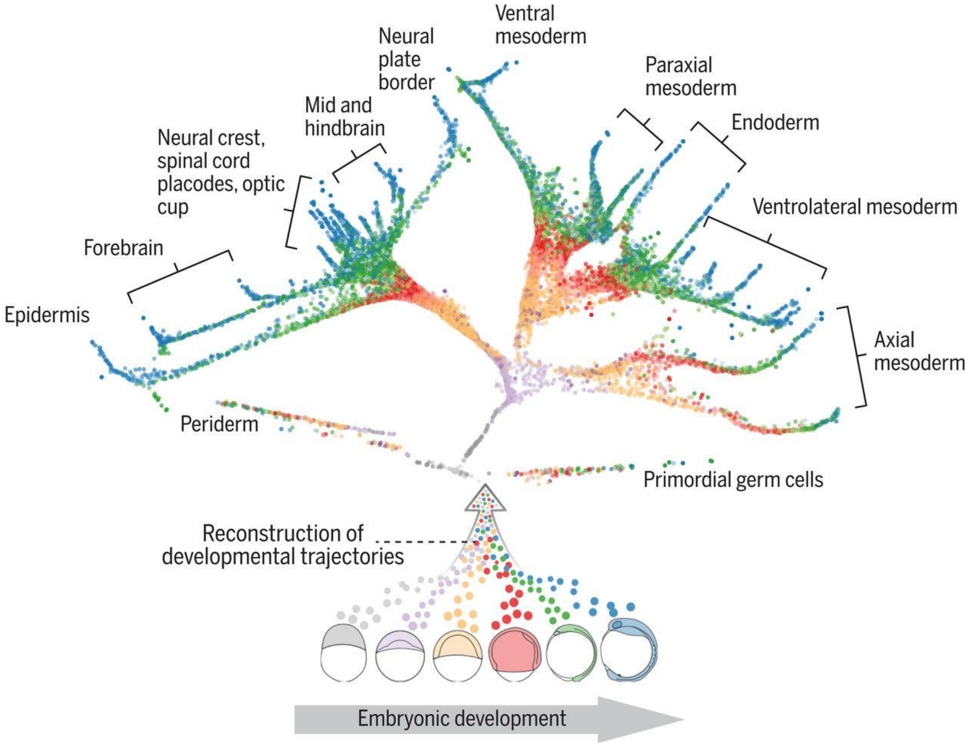


Figure 4. *The developmental tree of early zebrafish embryogenesis [15]. Transcriptomes were obtained from zebrafish embryos at 12 different developmental stages. The trajectories show a reconstruction of cell fates of 25 cell types. Each cell is represented by a point and colored by the developmental stage.*

Once cells have been ordered along a developmental trajectory, it becomes possible to investigate the gene expression patterns along the trajectory. This analysis allows us to identify key regulators and genes that are responsible for specific branches, exhibiting "switch-like" behavior.

ScRNAseq has made it possible to study coordinated changes in gene expression during dynamic biological processes, such as transcriptional kinetics [32]. The standard model for gene expression kinetics is a two-state model, where the transcription of a gene stochastically switches between "on" and "off" states. In other words, genes are not transcribed continuously but instead produce transcripts in intermittent bursts [33-36]. This phenomenon, known as “*transcriptional bursting*” can be investigated using allele-sensitive scRNAseq (**Figure 5**) [37].

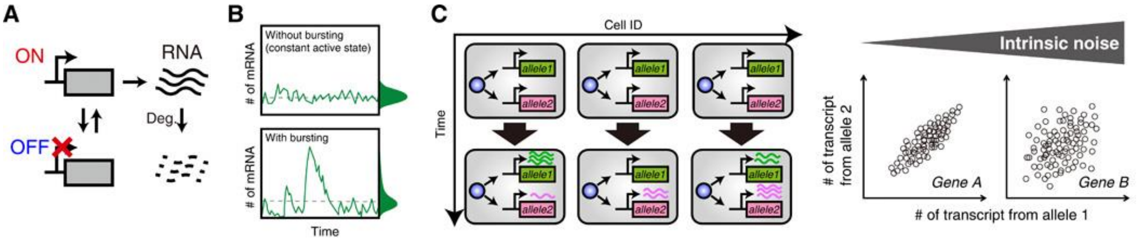

A
ON
OFF
RNA
Deg.
B
# of mRNA
Without bursting (constant active state)
With bursting
Time
C
Cell ID
Time
allele1
allele2
Intrinsic noise
# of transcript from allele 2
Gene A
Gene B
# of transcript from allele 1

Figure 5. scRNAseq for transcriptional kinetics studies [37]. (A) Schematic diagram of gene expression with stochastic switching between ON and OFF states. (B) Schematic representations of the dynamics of transcript levels of a gene with or without transcriptional bursting. (C) Transcriptional bursting induces inter-allelic and intercellular heterogeneity in gene expression (left). Scatter plots of the individual allele-derived transcript numbers (right)

Another area of research that can be explored through scRNAseq technology is gene regulatory network (GRN) analysis [38]. GRN analysis captures relationships between regulators of gene expression and their target genes. The concept is straightforward: if the product of gene A inhibits the expression of gene B, then their expression levels would be inversely correlated. Consequently, cells with high levels of gene A would exhibit low levels of gene B [39]. GRN analysis aims to identify functional gene modules that interact with each other under specific conditions, making it context-dependent. Jackson et al. [38], through GRN analysis, discovered novel regulatory connections and relationships related to nitrogen metabolism in *Saccharomyces cerevisiae* (**Figure 6**).

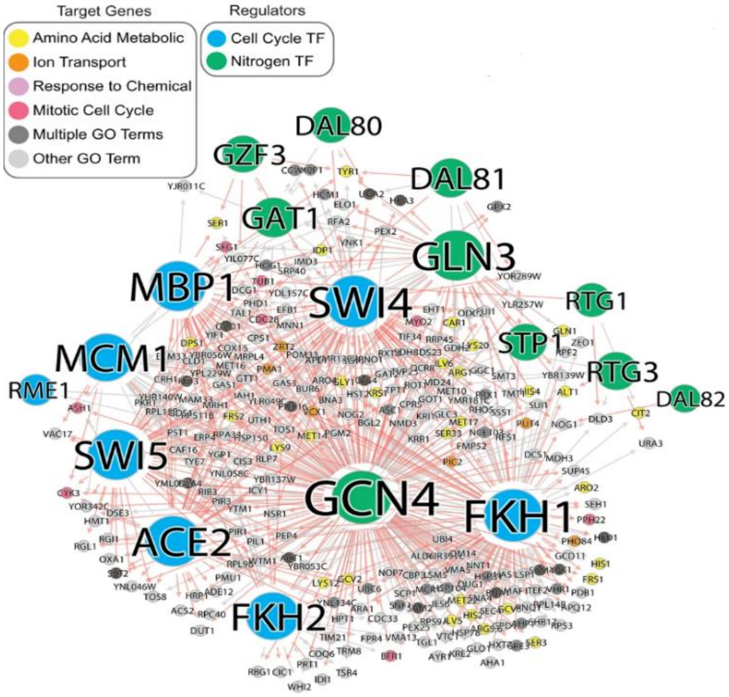


Figure 6. Gene regulatory network created by [38]. A GRN shows target genes that are regulated by at least one nitrogen TF (blue) and at least one cell cycle TF (green). Target gene nodes are colored by GO slim term. Newly inferred regulatory edges are red and known regulatory edges from the prior are in gray.

II.2 scRNAseq - experimental workflow

Single-cell RNA sequencing is a combination of high-yield cell separation techniques and next-generation sequencing (NGS). The experimental workflow typically includes the following steps:

1) single-cell isolation and mRNA extraction,

1) reverse transcription of mRNA to cDNA

2) cDNA pre-amplification,

3) library construction (cDNA fragmentation, barcoding, adaptor ligation)

4) sequencing

Capturing single cells from the whole tissue or cell sample is a crucial and most challenging step in the experiment. The main challenges associated with this step include throughput (the number of cells that can be isolated within a certain time), purity (the proportion of desired cells in the final isolated cell fraction), and recovery (the percentage of captured target cells from the starting sample) [40].

Several different cell isolation methods are available, which are applied based on the scientific objective. These methods utilize various cellular properties, such as size, density, or fluorescence [41]. Two general categories can be distinguished: plate-based methods (**Figure 7**) and microfluidics (**Figure 8**). Plate-based methods are considered low-throughput compared to microfluidics and are not suitable for the identification of rare cell types.

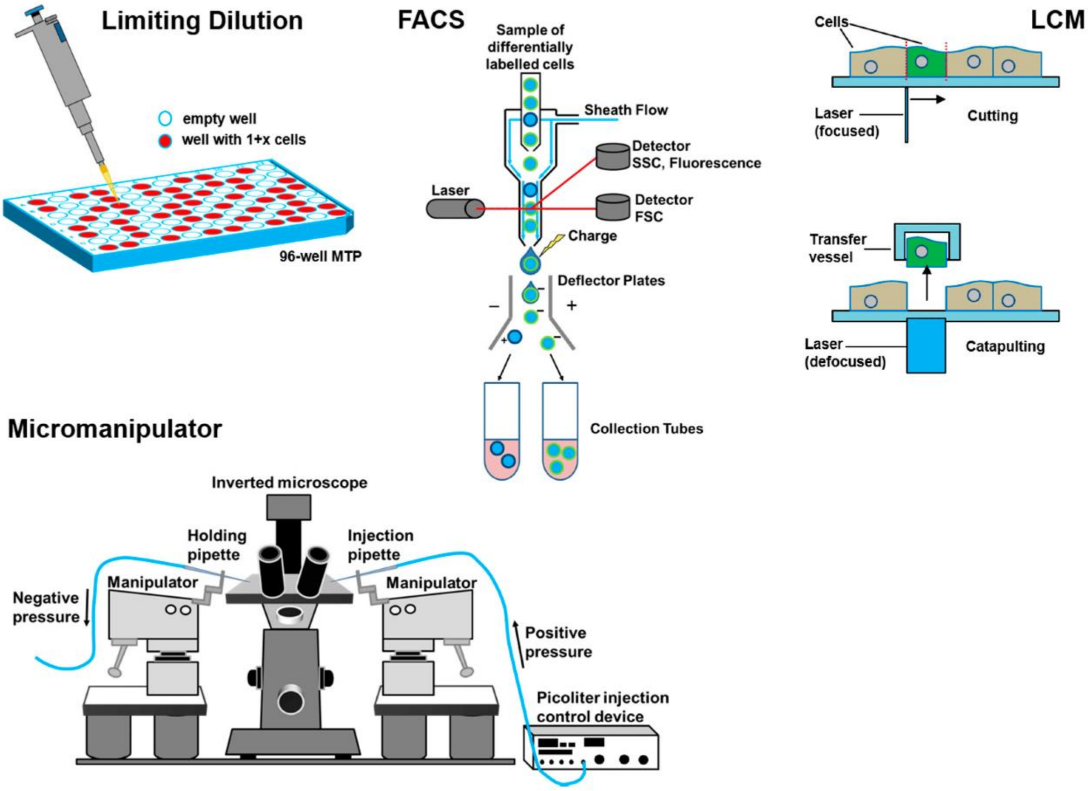


Figure 7. *Single-cell plate-based isolation methods [40]. (i) Limiting dilution method: the cell suspension is diluted to a point where only one cell is present in each microwell. It is a simple and cost-effective approach but lacks control and requires additional verification to confirm the presence of cells. (ii) Fluorescent activated cell sorting (FACS): target cells are labeled with fluorophore-conjugated monoclonal antibodies that recognize specific markers on the cells. When the cells pass through a laser beam, the fluorophore is excited, and the cells are selectively detected and sorted. FACS allows selection based on size and granularity but requires a large input of cells (more than 10,000 cells). Additionally, the viability of the sorted cells may be reduced due to the rapid flow in the machine. (iii) Laser capture microdissection (LCM): this method is suitable for solid tissue samples. A desired section of the sample is cut off using a laser beam, followed by the extraction of the isolated cell or compartment. LCM enables the selection of cells based on morphology and preserves spatial location information. However, it is time-consuming, expensive, and requires highly skilled personnel. (iv) Micromanipulator: this technique involves an inverted microscope combined with micro-pipettes for the manual isolation of cells, particularly embryo cells or live culture cells. It allows for processing only a small number of cells and requires a high level of skill.*

Microfluidic cell sorters utilize small channels with sizes ranging from approximately 100 nanometers to 500 micrometers to achieve precise control over fluid flow. These systems offer higher throughput compared to plate-based methods and enable the execution of all the necessary reactions for library preparation in nanoliter volumes. However, microfluidic-based methods are more prone to producing doublets, which means capturing two or more cells. The proportion of doublets can be minimized by loading cells at low concentrations, but this significantly increases the cost per cell. Commercially available microfluidic platforms typically operate based on three main principles to isolate individual cells (**Figure 8**) [40].

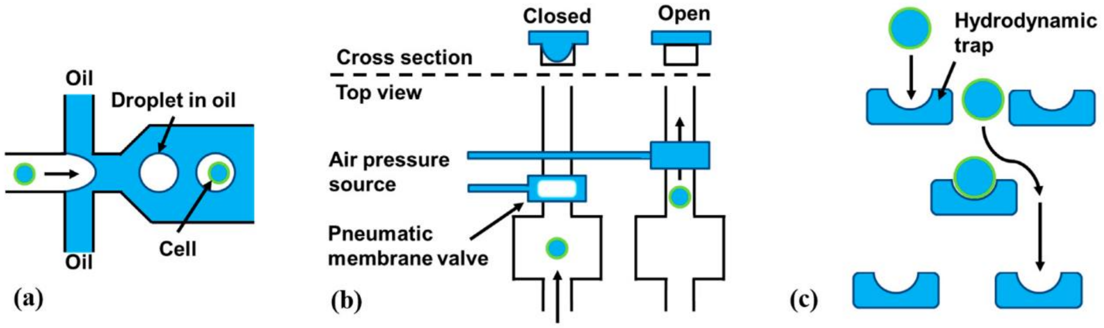


Figure 8. Main groups of microfluidic-based platforms [40]. (a) Droplet-in-oil: the cells are dispersed into individual droplets enclosed by oil in a random distribution, (b) Pneumatic membrane valves: work by using air pressure to deform a flexible membrane that separates the fluid channels, (c) Hydrodynamic cell traps: passive elements that permit the entry of only a single cell into the trap. This group of platforms is not suitable for cases where the number of cells is very limited, as only approximately 10% of cells can be captured (for example system C1 from Standard BioTools Inc).

Due to their cost-effectiveness and high throughput (with capture rates ranging from 60% to 90%), droplet-based platforms are currently the most widely used. Each droplet contains a resin bead and serves as an individual reaction chamber. The surface of each bead is coated with numerous oligonucleotide sequences that are important for sequencing and subsequent analysis (**Figure 9**) [42].

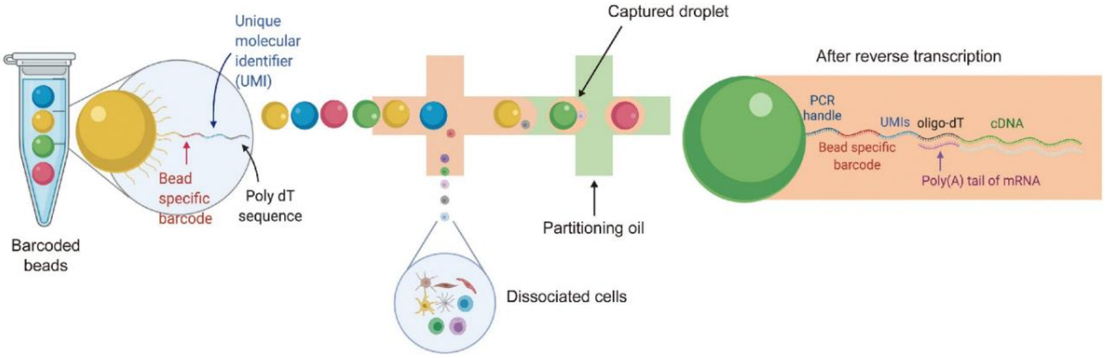


Figure 9. Droplet structure [42]. Each strand at the surface of a bead consists of four parts: (i) PCR handle: this part is a constant sequence that is identical for all beads. It functions as a priming site for downstream PCR and sequencing, (ii) Bead-specific barcode (cell barcode): this barcode is identical across all the primers on the surface of any given bead but differs from the cell barcodes on other beads. The cell bar-codes allow for the pooling of droplets and subsequent sequencing of them together, (iii) Unique molecular identifier (UMI): each oligo on the bead has a distinct UMI, (iv) Poly(T) tail at the 3' end: this tail is used to capture cellular mRNAs.

The choice of cell isolation method strongly depends on the origin of the input sample, which can be solid tissue, cell suspensions, or cell culture. Sample requirements also play a substantial role. In some cases, the enzymatic dissociation process can induce the expression of stress genes, resulting in artificial changes in cell transcription patterns [43, 44]. Compatibility with downstream applications is also an important consideration. For instance, the choice of a specific cell isolation method may impose limitations on the sequencing platform and the type of sequencing library that can be prepared [45].

Once the cells have been collected, they are lysed to allow the capture of as many RNA molecules (transcripts) as possible. The transcript-capturing rate varies depending on the protocol and chemistry used, ranging from 10% to 35%. It is important to note that the detection of a transcript is a random occurrence, and sequencing multiple single cells from the same population is necessary to capture the majority of the transcriptome.

RNA molecules are highly unstable and cannot be directly measured in the experiment. They need to be converted into a stable structure called complementary DNA (cDNA) through the process of reverse transcription (RT) or cDNA synthesis. The efficiency of this step is another crucial factor that affects the sensitivity of the scRNAseq experiment. It varies depending on experimental conditions and even transcripts of different genes [46][39]. A wide range of results has been reported in this regard [40] but on average, only 10-40% of transcripts are successfully reverse transcribed.

To be detected by the sequencer, cDNA needs to be duplicated (amplified) millions of times. The most commonly used method for amplification is the polymerase chain reaction (PCR), which utilizes exponential amplification. However, PCR is an imprecise process, and some transcripts are preferentially amplified (such as cDNA fragments of shorter length or lower GC content), while others may be amplified below their true expression level [47]. This leads to non-linear bias of some reads over others and the accumulation of nonspecific byproducts. An alternative method less prone to such bias is in vitro transcription (IVT) based on linear amplification. However, it requires more input cDNAs compared to PCR. Pre-amplified cDNAs undergo several steps of library preparation, including fragmentation, barcoding and adaptor ligation.

To overcome amplification bias, unique molecular identifiers (UMIs) are attached to each individual transcript within a cell during the reverse transcription step [48]. These short random sequences serve as molecular barcodes that enable the identification of individual RNA molecules [49]. Without UMIs, it would be impossible to differentiate PCR clones generated from identical fragments of the original transcript (**Figure 10**).

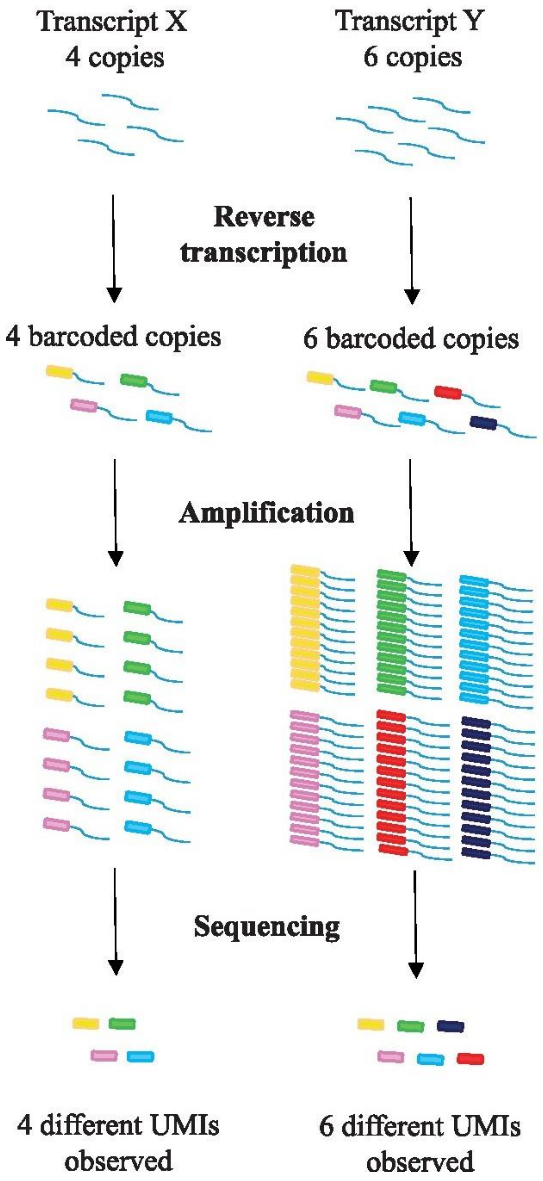


Figure 10. The principle of UMIs [48]. In this simplified hypothetical scenario, we have two transcripts, X and Y, which are expressed with four and six copies, respectively. To account for each individual copy, four different UMIs (represented by filled rectangles) are attached to the copies of transcript X, while six different UMIs are attached to the copies of transcript Y. Both transcripts undergo amplification, although with varying efficiency, with gene Y having a higher amplification efficiency. This may lead to the erroneous conclusion that transcript Y has higher expression compared to X. The number of true copies can be restored by considering UMIs which ensures that amplicons of the same read are only counted once. Subsequently, in downstream bio-informatic analysis, PCR clones are removed from the dataset.

The prepared libraries from each cell are subsequently pooled together (multiplexed) and loaded onto a flow cell for sequencing in a single run. The widely used next-generation sequencing (NGS) platform is Illumina, which employs the sequencing by synthesis (SBS) approach. In SBS, chemically modified nucleotides bind to the cDNA template strand through natural complementarity. Each nucleotide is tagged with a fluorescent marker and a reversible terminator, preventing the incorporation of the next base. The fluorescent signal indicates the added nucleotide, and upon removing the terminator, the next base can be bound. This process is repeated for both the forward and reverse DNA strands, known as paired-end sequencing.

During sequencing, cDNA fragments are randomly captured or "sampled," resulting in a vast number of short reads (**Figure 11**). These reads are subsequently computationally aligned to a reference genome to annotate each transcript with its corresponding gene name and cell of origin.

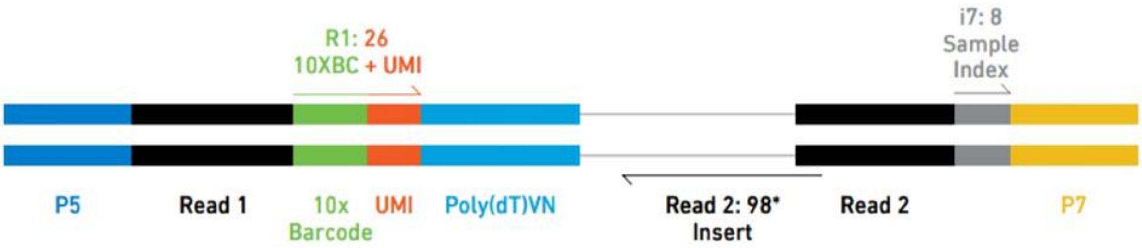
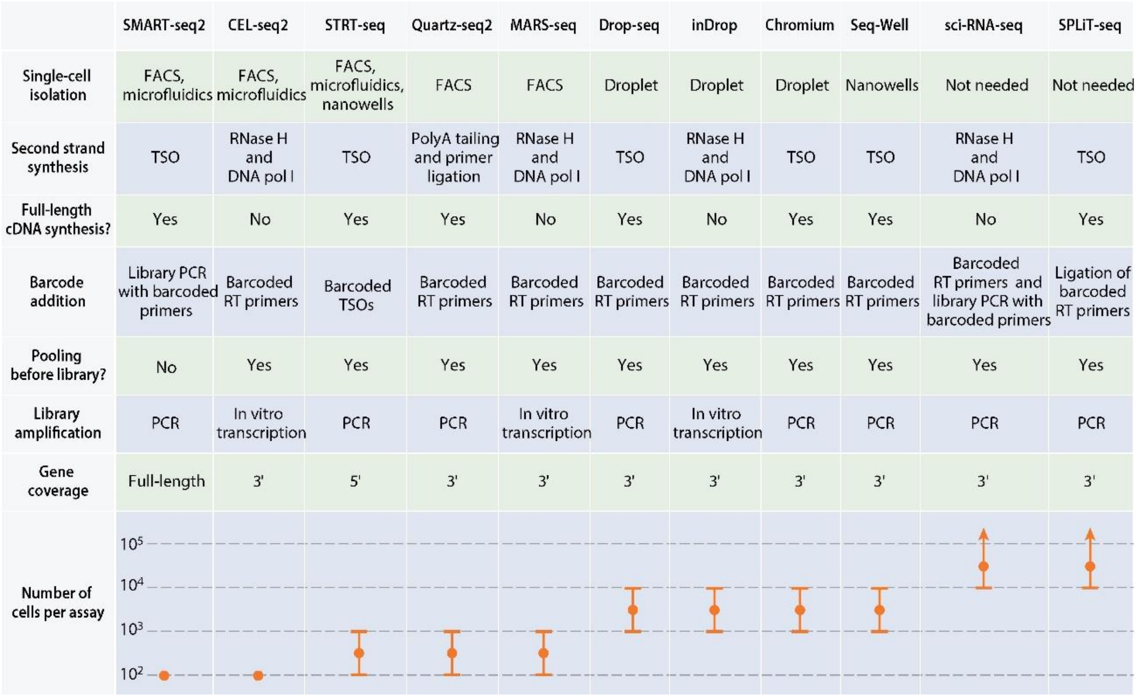


Figure 11. *Schematic of a read fragment from Chromium™ Single Cell 3' v2 library. Each read consists of the following elements: (i) sample index: determines which sample the read originated from, (ii) 10x Barcode (cellular barcode): determines which cell the read originated from, (iii) UMI barcode: determines which gene transcript the read originated from, (iv) Read1: cellular + UMI barcode, (v) Read2: transcript sequence.*

There are two methods used for measuring expressed transcripts in cells: tag-based and full-length protocols. In tag-based protocols, only a short fragment (tag) of the transcript located at one end (3' or 5') is sequenced. However, due to this restriction, the ability to align reads unambiguously to a reference is diminished. Additionally, the complete information about transcript structure is lost, making it challenging to differentiate between various transcript isoforms (splice variants) [50]. On the other hand, sequencing transcripts in their entirety (full-length protocols) allows for a more comprehensive characterization of the internal transcriptional state of cells and enables the detection of alternative splice variants of genes. However, full-length methodologies often introduce a bias towards long genes, as long transcripts tend to have a higher number of reads mapped to them compared to short genes with similar expression levels [51, 52].

Many different scRNAseq protocols have been developed (**Figure 12**) [53]. for example MARS-seq [54], Smart-seq2 [55] or Chromium [56]. These protocols incorporate different optimizations to enhance cell yield and viability, improve the efficiency of reverse transcription (RT), and enhance transcript quantification . Detailed comparisons among various protocols are available in publications such as [53, 57-59]. Summarizing these evaluations, significant differences are observed between protocols in terms of their sensitivity in capturing RNA molecules (i.e., the number of detected genes) and accuracy, as measured by Pearson's correlation with bulk RNAseq data. Furthermore, variable performance has been reported in their ability to distinguish between different cell types.



	SMART-seq2	CEL-seq2	STRT-seq	Quartz-seq2	MARS-seq	Drop-seq	inDrop	Chromium	Seq-Well	sci-RNA-seq	SPLiT-seq
Single-cell isolation	FACS, microfluidics	FACS, microfluidics	FACS, microfluidics, nanowells	FACS	FACS	Droplet	Droplet	Droplet	Nanowells	Not needed	Not needed
Second strand synthesis	TSO	RNase H and DNA pol I	TSO	PolyA tailing and primer ligation	RNase H and DNA pol I	TSO	RNase H and DNA pol I	TSO	TSO	RNase H and DNA pol I	TSO
Full-length cDNA synthesis?	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	No	Yes
Barcode addition	Library PCR with barcoded primers	Barcoded RT primers	Barcoded TSOs	Barcoded RT primers	Barcoded RT primers	Barcoded RT primers	Barcoded RT primers	Barcoded RT primers	Barcoded RT primers	Barcoded RT primers and library PCR with barcoded primers	Ligation of barcoded RT primers
Pooling before library?	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Library amplification	PCR	In vitro transcription	PCR	PCR	In vitro transcription	PCR	In vitro transcription	PCR	PCR	PCR	PCR
Gene coverage	Full-length	3'	5'	3'	3'	3'	3'	3'	3'	3'	3'

Figure 12. Comparison of common scRNA-seq protocols [53]. Abbreviations: cDNA - complementary DNA; DNA pol I - DNA polymerase I; FACS - fluorescence-activated cell sorting; PCR - polymerase chain reaction; RNase H - ribonuclease H; RT - reverse transcription; TSO - template-switching oligonucleotide

II.3 Noise in scRNAseq data

Single-cell RNA sequencing data exhibits a high level of measurement noise, which arises from both technical and biological sources of variation (**Figure 13**) [60]

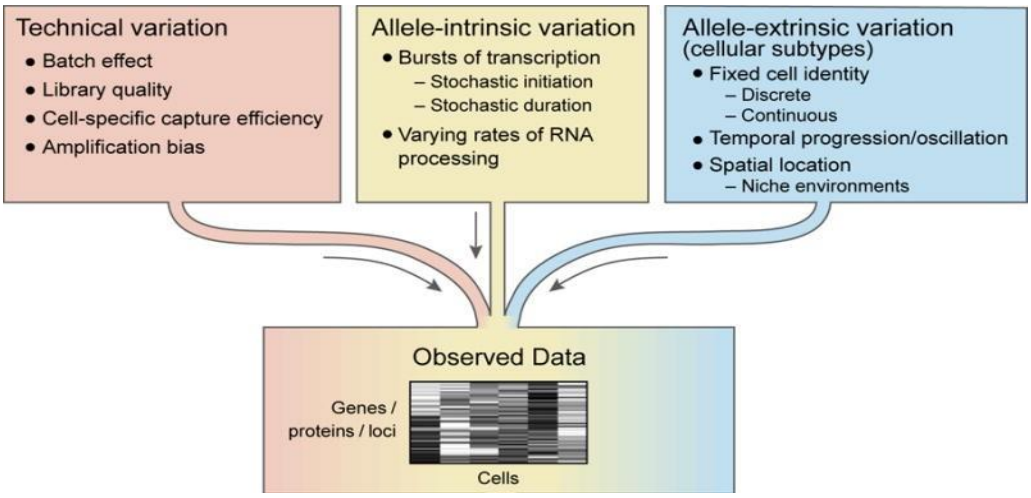


Figure 13*. Sources of variation in scRNAseq data [60]. Biological and technical factors contribute to the observed gene expression profiles of single cells. Biological factors are divided into allele-intrinsic group reflecting stochastic fluctuations that do not correlate between two alleles of the same gene, and allele-extrinsic factors that are related with different cell types and states.*

Technical factors are associated with variations during library preparation, particularly in sample handling and processing. Some cells may be missed or lost during the isolation or sorting step. Storage, or processing conditions can impact the quality and integrity of the RNA. However, the main source of noise is related to imperfect measurement process of minute amounts of input mRNA obtained from individual cells, typically on the scale of picograms. Variations in cell lysis efficiency and cDNA conversion as well as PCR amplification bias lead to differences in total number of reads (sequencing depth) across cells. Additionally, different sequencing platforms have their own limitations in terms of read capacity or throughput. Some platforms may generate a higher number of reads per run or have longer read lengths, allowing for higher sequencing depth, while others may have lower capacities. A large group of technical factors that make substantial contributions to the overall noise in scRNAseq data is collectively termed batch effects. This group will be discussed in the next section.

Many of the aforementioned factors, are addressed through normalization procedures, but others, such as batch effects, require dedicated approaches [61-63]. It is important to note that UMI barcodes are specifically designed to capture amplification bias and are unable to account for differences in capture efficiency before the reverse transcription [64].

Examples of the variables that contribute to cellular heterogeneity and shape the biological context of a cell involve transcriptional bursting, the cell cycle, developmental stage, and tissue landscape (**Figure 14**).

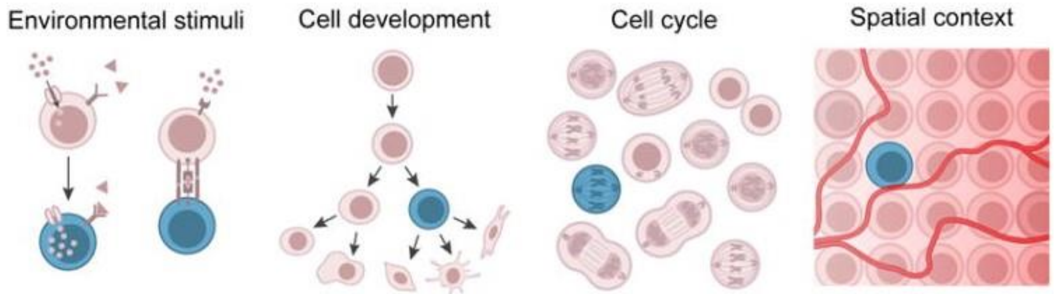
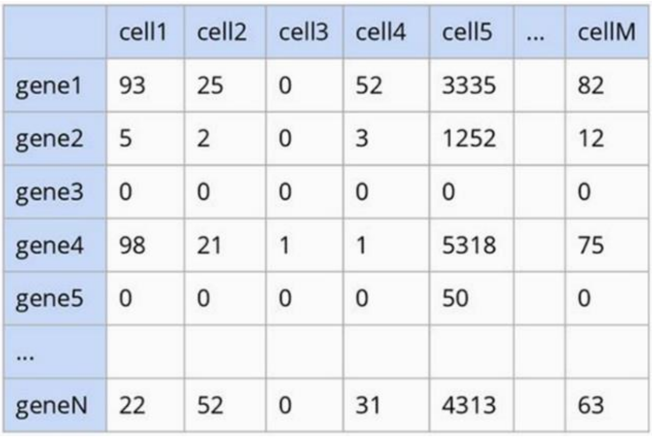


Figure 14. *Biological contexts of a cell [60]. A particular cell (highlighted in blue) experiences multiple concurrent contexts that shape its identity simultaneously. These are: (i) environmental stimuli like nutrient availability or signaling molecule binding, (ii) developmental stage, (iii) cell cycle, and (iv) spatial context (tissue landscape) that determines oxygen availability, cellular neighbors, and developmental cues, such as morphogen gradients.*

Depending on the biological question at hand, certain biological sources of variation, such as transcriptional bursting, may either be a focal point of the experiment or considered a nuisance factor.

II.3.1 Zero measurements in scRNAseq data

High proportions of genes with zero expression measurements are commonly observed in scRNAseq data (**Figure 15**).

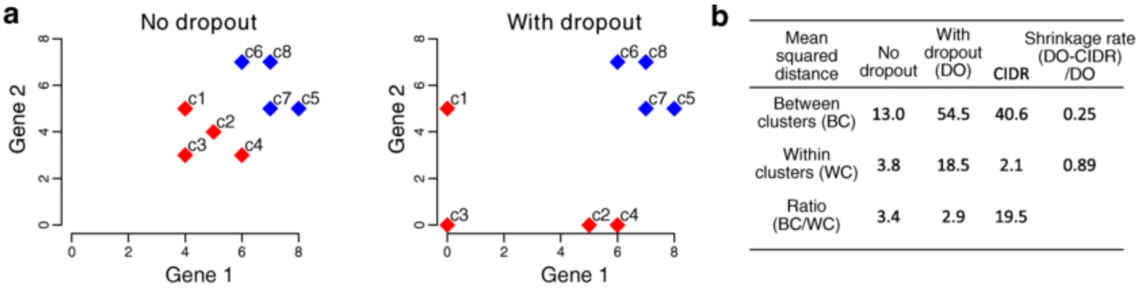


	cell1	cell2	cell3	cell4	cell5	...	cellM
gene1	93	25	0	52	3335		82
gene2	5	2	0	3	1252		12
gene3	0	0	0	0	0		0
gene4	98	21	1	1	5318		75
gene5	0	0	0	0	50		0
...							
geneN	22	52	0	31	4313		63

Figure 15. *An example of scRNAseq expression matrix*

Different terms such as "dropouts," "excess zeros," or "zero inflation" are used in the literature to describe the prevalence of zeros in scRNAseq data [65]. This inconsistency in terminology has significant implications for researchers, particularly in the development and application of analysis methods [65, 66].

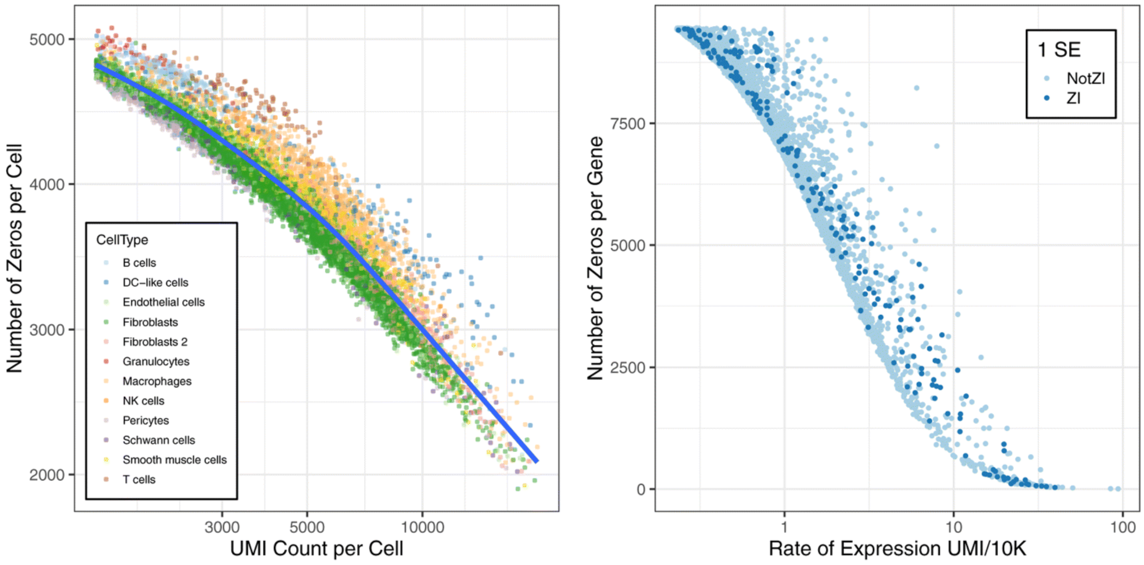
The frequency of zeros, also known as the dropout rate, indicates the level of data sparsity and often exceeds 50% of all entries in the expression matrix [67, 68]. Dropouts can have an impact on clustering analyses by increasing cell-to-cell dissimilarity, potentially leading to misclassification of cell types [69]. In other words, they distort the relative positions of cells in the low-dimensional subspace (**Figure 16**) [70].



Mean squared distance	No dropout	With dropout (DO)	CIDR	Shrinkage rate (DO-CIDR) /DO
Between clusters (BC)	13.0	54.5	40.6	0.25
Within clusters (WC)	3.8	18.5	2.1	0.89
Ratio (BC/WC)	3.4	2.9	19.5	

Figure 16*. The effect of dropouts on clustering scRNAseq data (toy example) [70]. a – eight cells are divided into two clusters (the red and blue). b - dropouts cause a significant increase in the within-cluster distances among the single cells in the red cluster, as well as an increase in between-cluster distances between single cells in the two clusters*

Empirical observations regarding dropouts indicate that sequencing depth (total number of UMIs per cell) is the primary factor influencing dropout rates [71]. The relationship between sequencing depth and dropout rate is inverse, meaning that higher sequencing depth results in lower dropout rates (**Figure 17a**). When the sequencing depth is low, fewer reads are obtained from each individual cell, making genes with lower expression levels more vulnerable to stochastic fluctuations and measurement noise. Consequently, such genes are more likely to be affected by dropouts compared to genes with higher expression magnitudes (**Figure 17b**) [69]. Furthermore, shorter genes are more susceptible to dropout events due to their limited number of regions available for RNA transcription and detection during the sequencing process [51].



(a) (b)

Figure 17. Factors influencing dropout rates [69]. (a) – higher the sequencing depth, the higher dropout rate. Color cod-ing indicates cell types, (b) – relationship between average gene expression and the number of zeros per gene. Genes identified as zero-inflated (ZI) are indicated in dark blue.

Zero measurements in scRNAseq data can stem from both technical and biological factors, as illustrated in **Figure 18** [65]. A biological zero occurs when a gene transcript is genuinely absent in a cell, providing valuable information about the cell's transcriptional state. These biological zeros primarily arise due to the aforementioned transcriptional bursting phenomenon. Technical zeros are artificially introduced during the data generation process. This involves situations when a transcript of a gene may be present in a cell but not captured (missed) during the conversion to cDNA. As a result, it would not be detected during sequencing. Such zeros are often termed as purely technical, instead of sampling zeros arising from limited efficiency of amplification, cDNA conversion and sequencing depth [65].

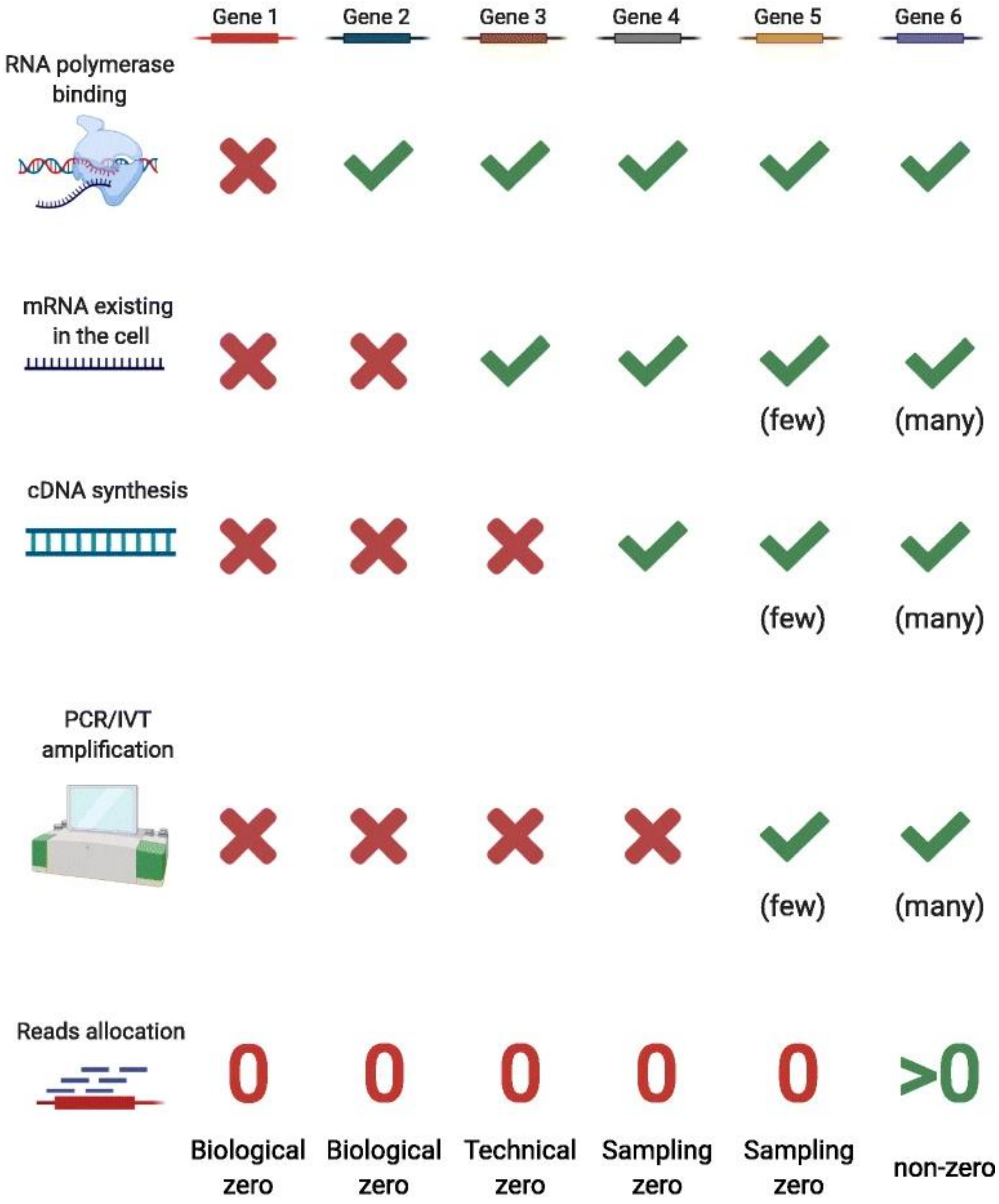


Figure 18. *Sources of zeros in scRNA-seq data [65]. Red crosses indicate occurrences of zeros, while green checkmarks indicate otherwise. Biological zeros arise from two scenarios: no transcription (gene 1) or no mRNA due to faster mRNA degradation than transcription (gene 2). If a gene has mRNAs in a cell, but its mRNAs are not captured by cDNA synthesis, the gene's zero expression measurement is called a technical zero (gene 3). If a gene has cDNAs in the sequencing library, but its cDNAs are too few to be captured by sequencing, the gene's zero expression measurement is called a sampling zero. Sampling zeros occur for two rea-sons: a gene's cDNAs have few copies because they are not amplified by PCR or IVT (gene 4), or a gene's mRNA copy number is too small so that its cDNAs still have few copies after amplification (gene 5). If the factors and procedures above do not result in few cDNAs of a gene in the sequencing library, the gene would have a non-zero measurement (gene 6).*

Zeros in scRNAseq data are perceived differently by researchers. Some studies attribute them to artifacts, either technical [68, 72] or biological [71] in nature. These studies suggest utilizing statistical models that incorporate covariates accounting for sparsity, sampling variation, and other types of noise [71, 73]. Zero-inflated models are commonly used for this purpose, although research conducted by Svensson [74] suggests that they may not be suitable for UMI-based scRNAseq data, as he demonstrates that such data is not actually zero-inflated.

On the other hand, proponents of treating zero measurements as missing data aim to develop tools for the data imputation [75-78]. Some tools, such as SAVER [77] or scImpute [79] are based on probabilistic models that identify technical zeros and impute expression values only for these ‘false’ zeros, while leaving ‘true’ ones unaltered. However, these methods assume homogeneous cell populations, raising concerns about their suitability for identifying novel rare cell types Other methods, like DrImpute [76] or MAGIC [75], are data smoothing-based approaches that correct the entire expression matrix, including both technical and biological zeros, as well as non-zero values. However, this approach may eliminate meaningful biological variation.

Although imputation algorithms may improve certain analyses such as dimensionality reduction, visualization, or clustering, their performance depends on various factors, including the experimental protocol, data sparsity, the number of cells in the dataset, and the effect size between differentially expressed genes [80]. Most methods rely solely on the data to be imputed, leading to a circularity problem where biases present in the dataset, including random noise or confounding signals, can be amplified. This can introduce false positives in downstream analyses, such as differential gene expression and gene network inference [81]. Moreover, the majority of dropout imputation methods do not provide uncertainty quantification.

In between these two opposing camps, there are authors who recognize the potential for leveraging dropouts to identify cell types [82, 83] or for feature selection [67]. The former indicates that cellular heterogeneity can drive excessive zeros. Consequently, these studies conclude that zero proportions can serve as a metric for distinguishing between various cell types. However, further research is required to validate these approaches across diverse scRNAseq datasets and experimental settings.

II.3.2 Batch effects in scRNAseq data

One of the most challenging sources of unwanted variations in scRNAseq data is known as "batch effect." Broadly speaking, this term refers to variability caused by factors unrelated to the specific biological variables being investigated, such as disease severity, cancer type, gender, and so on. Batch effects arise from the experimental design and handling procedures when different biological groups of interest are processed separately or differently, in what we refer to as batches [84]. These variations can stem from differences in technology platforms, reagent lots, experiment execution times, and handling personnel, all of which introduce variability that confounds the biological variations of interest. It is worth noting that confounding factors may not even be documented during the course of the experiment. In extreme cases (completely confounded design), batch effects may be the major drivers of heterogeneity, which means that batch explains more variability than the biological group. In such scenario, it is impossible to attribute the observed variation in the data to biology or batch effects (**Figure 19**) [84]. Batch effects obscure the biological variation of interest in a manner that is not fully understood, resulting in false discoveries and misleading interpretations of the data. These effects pose a particular challenge for the analysis of scRNAseq data, as it is commonly based on unsupervised learning methods. While an appropriate experimental design incorporating blocking, randomization, and the use of replicates can help minimize the negative impact of batch effects, achieving a perfectly balanced experimental design is not always feasible, especially in large projects where data generation must be carried out separately due to logistical constraints. In such cases, it becomes necessary to employ computational approaches for batch effect correction or removal. Additionally, batch effects can be accounted for by including batch variables (if recorded) as covariates (additional predictors) in the statistical model [85, 86]. The underlying assumption of these methods is that composition of cell populations across batches is the same across batches. However, in scRNAseq data analysis such a priori knowledge is usually unavailable.

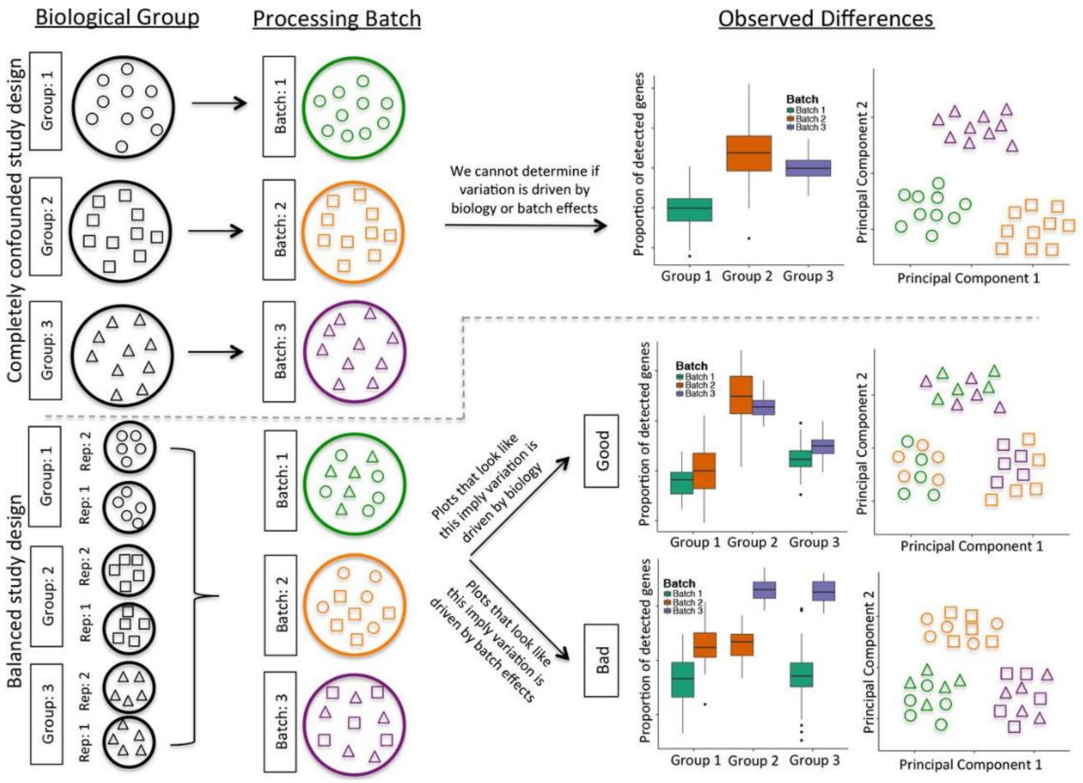

Biological Group
Processing Batch
Observed Differences
Completely confounded study design
Group: 1
Group: 2
Group: 3
Batch: 1
Batch: 2
Batch: 3
We cannot determine if variation is driven by biology or batch effects
Proportion of detected genes
Batch
Batch 1
Batch 2
Batch 3
Group 1
Group 2
Group 3
Principal Component 2
Principal Component 1
Balanced study design
Rep: 1
Rep: 2
Plots that look like this imply variation is driven by biology
Good
Plots that look like this imply variation is driven by batch effects
Bad

Figure 19. *Source of batch effect in data [84]. Completely confounded study design is shown in top section where cells from three biological groups (represented by shapes) are processed as three different batches (rep-resented by colors). In this scenario, we cannot identify if biology or batch effects drive the observed variation (cells from each batch cluster together). A balanced study design corresponds to such scenario where cells from a biological group are split and processed in multiple replicates (rep) across different batches. Such design allows observed variation to be attributed to biology (cells cluster by shape) or batch effects (cells cluster by color)*

The goal of batch effects correction in scRNAseq is to 'force' cells of similar types to cluster/group together. These cells should be intermingled and indistinguishable even if they originate from different batches. In other words, datasets should be integrated to be jointly analysed. Batch correction is essential for facilitating data integration across multiple omics modalities in single cells or for cross-species comparisons.

However, accomplishing this task is not straightforward due to several challenges:

- Complex data structure: scRNA-seq data is high-dimensional, consisting of expression measurements from tens of thousands of genes across hundreds of thousands of individual cells. Additionally, scRNA-seq data is highly sparse, which makes it difficult to differentiate true biological variability from technical variability and often leads to overcorrection.
- Nonlinear and nonadditive nature of batch effects: batch effects can be highly nonlinear and context specific as well as they can exhibit high complexity (nested layers of unwanted variation) making them difficult to model.

- Unbalanced batch sizes: some methods may prioritize larger batches, leading to biased results.
- Different composition of cell populations across batches.
- High heterogeneity of cells within the same cell type: even cells of the same type can have different gene expression profiles.
- Necessity of optimization: many current algorithms utilize intricate models involving several parameters that require tuning for individual datasets. For instance, parameters such as the number of nodes in each hidden layer of a neural network or the learning rate during training may lack clear biological or statistical interpretations. Moreover, improper parameter tuning can lead to significant performance degradation.

It is important to note that batch effect correction is distinct from data normalization procedure. Both concepts aim to correct for unwanted technical variation. However, normalization focuses on addressing systematic biases generated within each sequencing experiment, whereas batch correction targets the bias generated across batches. In other words, batch effects cannot be fully addressed by normalization, as they may affect different genes in different ways [87].

Data integration represents a highly active area of development in the analysis of single-cell genomics data, with over 200 tools currently available for this purpose (**Figure 20**).

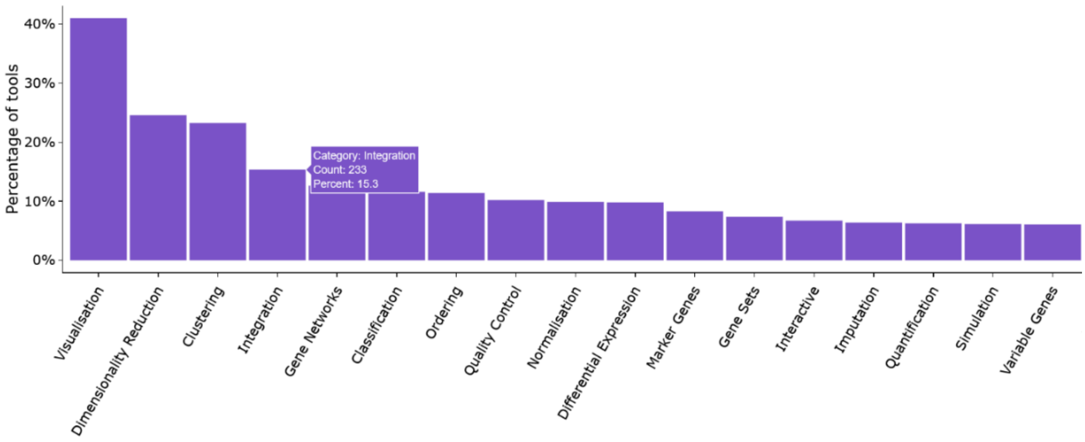


Figure 20. *Percentage of tools for different areas of analysis of scRNAseq data (as of May 2023)*

Different algorithms have been developed to address batch effects in scRNAseq data, employing various approaches (**Table 1**) [88].

Table 1. *Examples of batch correction methods [88].*

Tool	Programming language	Output type	Concept
Seurat 2	R	Embeddings	Canonical correlation analysis
Harmony	R	Embeddings	Iterative clustering in dimensionally reduced space
fastMNN	R	Embeddings	Mutual nearest neighbors in dimensionally reduced space
MND-ResNet	Python	Embeddings	Machine learning
LIGER	R	Embeddings	Matrix factorization
BBKNN	Python/R	Embeddings	k-nearest neighbors
Seurat 3	R	Normalized gene expression matrix	Canonical correlation analysis and mutual nearest neighbors
MNN Correct	R	Normalized gene expression matrix	Mutual nearest neighbors in gene expression space
ComBat	R	Normalized gene expression matrix	Linear models
limma	R	Normalized gene expression matrix	Linear models
scGen	Python	Normalized gene expression matrix	Machine learning
scMerge	R	Normalized gene expression matrix	Mutual nearest clusters
ZINB-WaVE	R	Normalized gene expression matrix Embeddings	Model-based
Scanorama	Python/R	Normalized gene expression matrix/ Embeddings	Mutual nearest neighbors in gene expression space

Many of these methods utilize the concept of mutual nearest neighbor (MNN), for instance MNN Correct [89], Scanorama [90] or scMerge [91]. They were designed to overcome the limitations of previous generation methods that relied on linear models and assumed identical compositions of cell population across batches. MNN-based methods require the presence of at least one shared cell population across batches. Mutual nearest neighbors, also known as "anchors," are pairs of cells that exhibit similar expression patterns across different batches. The process of identifying MNN pairs is illustrated in **Figure 21**. However, to align these pairs accurately, MNN methods make a rather strong assumption that true biological differences are nearly orthogonal or uncorrelated to those attributed to batch effects. It is important to note that this orthogonality assumption may not always hold in real-world datasets.

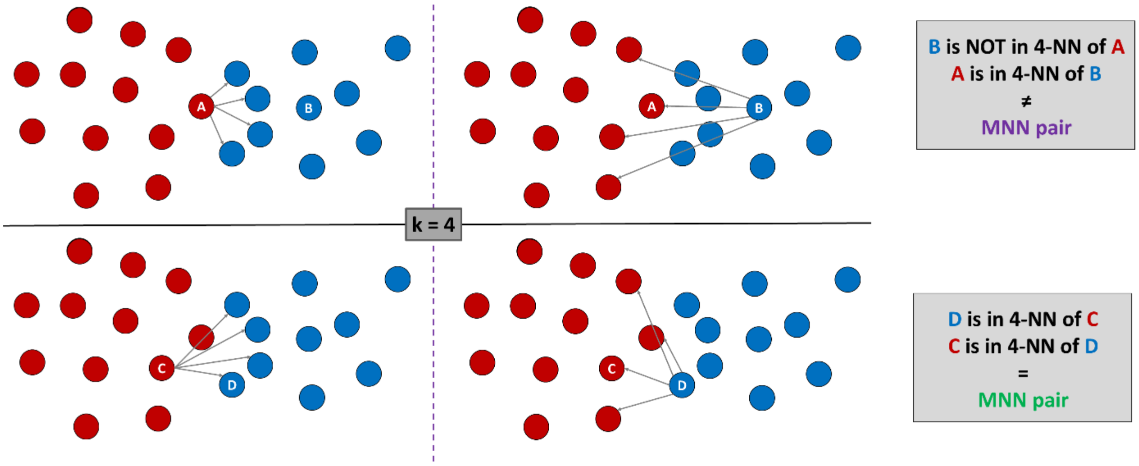
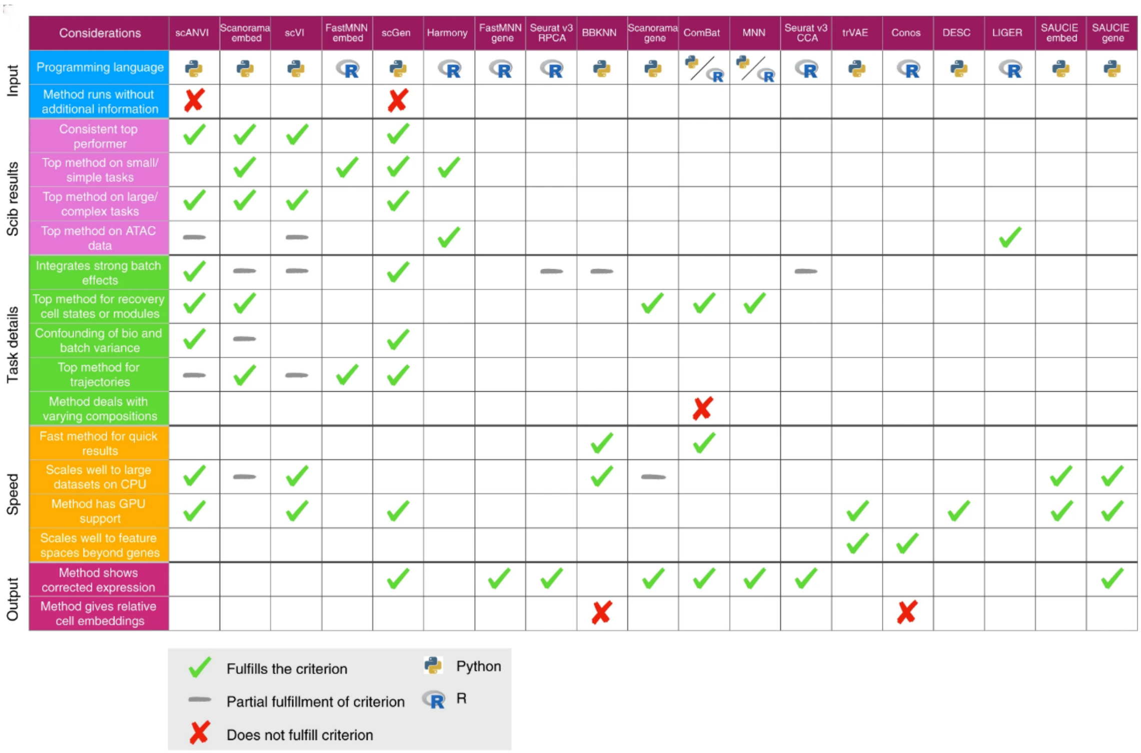


Figure 21. *Idea behind anchor-based methods (mutual nearest neighbors). k - the number of nearest neighbors to consider when identifying MNNs*

When deciding on the appropriate algorithm, several factors need to be considered, including the output generated by the method and whether it operates in a supervised or unsupervised manner. Certain methods, like Harmony [92] do not produce corrected data in the original space (normalized gene expression matrix). Instead, they generate dimension-reduced outputs, often referred to as canonical components or feature reduction vectors. However, such outputs are not compatible with downstream feature-level analyses such as differential expression or trajectory inference, limiting their applicability. The second crucial factor to consider is whether we have cell type labels or information about the cell types being analysed. If this information is not available, it becomes necessary to opt for unsupervised tools that do not rely on predefined cell type labels for correction. Deciding on the appropriate choice for a "batch" variable is not as straightforward as it may appear. The most common approach is to designate each sample as a separate batch, which typically results in a strong correction. However, in scRNAseq samples are usually confounded with biological factors of interest that should be preserved. To illustrate, let's consider an experiment where samples are collected at two different time points after drug administration. If these samples are treated as separate batches, the correction process will aim to remove the differences between the batches, inadvertently removing the underlying biological differences between the time points themselves.

Batch effects correction and clustering are interrelated tasks, hence all cell-level analyses, such as trajectory inference, are generally safe to perform on corrected data since correction algorithms aim to place cells in the same coordinate space. However, the same cannot be said for feature-level analyses, as correction algorithms are not obliged to preserve relative differences between individual genes. Moreover, our own research has demonstrated that many algorithms distort the original count-based nature and distribution of the data [93]. Additionally, certain methods like MNN or Scanorama introduce negative values in the corrected matrix, which can be challenging to interpret from a biological standpoint. This may result in artificial differential expression between cell types or experimental conditions. Furthermore, due to the loss of several properties of the original data after correction, downstream analysis tools based on models may not perform optimally with the corrected data.

Two comprehensive evaluations of the algorithms for batch effect correction in scRNAseq data were recently published [88, 94]. These studies examined numerous state-of-the-art methods based on their ability to remove batch effects while preserving biological variation, computational runtime, and scalability to large datasets. Different scenarios, including diverse cell types, multiple batches, nested batch effects, and simulated data, were investigated. Tran et.al demonstrated improved recovery of differentially expressed genes (DEGs) using ComBat, limma, and MNN, however in simulation scenario. Luecken et al. showed that integration methods may inadvertently remove biological variation along with the batch effect. Both studies provide recommendations for method selection, but no clearly superior method has emerged (**Figure 22**) [94]. The performance of batch correction algorithms depends significantly on the specific characteristics and complexity of the datasets and batch effects involved as well as the batch order in which the correction is applied.



	Considerations	scANVI	Scanorama embed	scVI	FastMNN embed	scGen	Harmony	FastMNN gene	Seurat v3 RPCA	BBKNN	Scanorama gene	ComBat	MNN	Seurat v3 CCA	trVAE	Conos	DESC	LIGER	SAUCIE embed	SAUCIE gene
Input	Programming language	🐍	🐍	🐍	R	🐍	R	R	R	🐍	🐍	🐍/R	🐍/R	R	🐍	R	🐍	R	🐍	🐍
	Method runs without additional information	✗				✗														
Scib results	Consistent top performer	✓	✓	✓		✓														
	Top method on small/ simple tasks		✓		✓	✓	✓													
	Top method on large/ complex tasks	✓	✓	✓		✓														
	Top method on ATAC data	–		–			✓											✓		
Task details	Integrates strong batch effects	✓	–	–		✓			–	–				–						
	Top method for recovery cell states or modules	✓	✓								✓	✓	✓							
	Confounding of bio and batch variance	✓	–			✓														
	Top method for trajectories	–	✓	–	✓	✓														
	Method deals with varying compositions											✗								
Speed	Fast method for quick results									✓		✓								
	Scales well to large datasets on CPU	✓	–	✓						✓	–								✓	✓
	Method has GPU support	✓		✓		✓									✓		✓		✓	✓
	Scales well to feature spaces beyond genes														✓	✓				
Output	Method shows corrected expression					✓		✓	✓		✓	✓	✓	✓						✓
	Method gives relative cell embeddings									✗						✗				

✓ Fulfills the criterion

– Partial fulfillment of criterion

✗ Does not fulfill criterion

🐍 Python

R R

Figure 22. Guidelines to choose batch correction algorithm [94]. The methods that do not fulfill a criterion are highlight-ed with a cross (only when a criterion is fulfilled by more than half of the methods The evaluation is con-ducted within the framework of scIB, which stands for single cell Integration Benchmark.

II.4 General workflow of scRNAseq data analysis

The general workflow of scRNAseq data analysis is depicted on **Figure 23** [48]. This workflow is applicable to all scRNAseq datasets, although there are subtle differences in the analysis between scRNAseq protocols. However, presenting a detailed explanation of these differences is beyond the scope of this thesis. The typical workflow consists of two main successive stages:

- Data preprocessing: crucial for transforming raw sequencing data into a format suitable for downstream analyses which is an expression matrix.
- Data cleaning: in this stage, general analyses are applied to reduce improve signal-to-noise ratio and address common issues that arise in scRNAseq datasets.
- Biological exploratory analysis: this stage is highly dependent on the specific research scenario or experimental design.

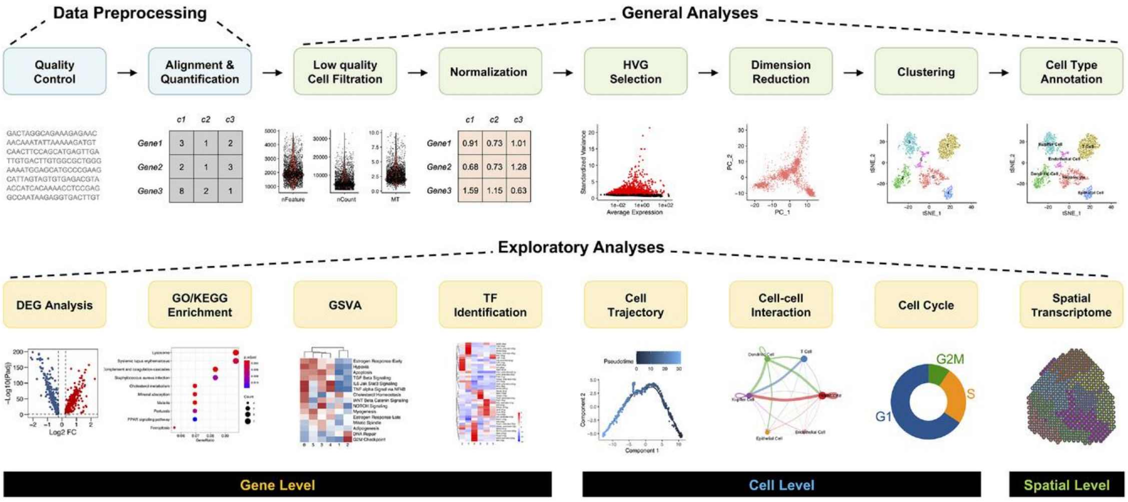


Figure 23. General pipeline for scRNAseq data analysis [48]. Three main steps can be distinguished: data preprocessing (blue panel), general analyses (green panel) and exploratory analyses (yellow panel). The plot below each box gives a schematic of the visualized results in each analysis step. HVG - highly variable gene; DEG - differentially expressed gene; GO – Gene Ontology; KEGG - Kyoto Encyclopedia of Genes and Genomes; GSVA - gene set variation analysis; TF - transcription factor.

It is worth mentioning that while the workflow outlined above is widely used, it is not considered a gold-standard analysis pipeline. The field of scRNAseq data analysis is continually evolving, and researchers often tailor their approaches to match the unique characteristics of their datasets and research goals. Therefore, it is crucial to adapt the analysis pipeline accordingly to ensure accurate and meaningful interpretation of scRNAseq data.

Data preprocessing

Preprocessing aims to generate the expression matrix, which can take the form of a count matrix or a read matrix depending on whether unique molecular identifiers (UMIs) were utilized [95]. The raw sequencing data obtained from the sequencing facility undergoes a series of steps, including quality control, read formatting, sample demultiplexing, genome alignment, and transcript quantification. There are two approaches to perform preprocessing:

- Combining individual methods: researchers can choose to combine specific methods tailored for each step [96-99]. This approach provides more control over each preprocessing step, allowing for customization based on specific requirements.
- Using available pipelines: such as CellRanger [56] or scPipe [100], which provide comprehensive preprocessing capabilities and streamline the entire process, however at the cost of control.

Low quality cell filtration

The expression matrix contains a significant amount of measurement noise, making it necessary to undergo several steps of data cleaning. The first crucial step is quality control at the cell level, which aims to identify and filter out cells that are dead, broken, or affected by technical issues during library preparation, such as doublets or empty droplets. This is achieved by analysing various metrics, such as the library size, the number of detected genes per cell, and the proportion of reads mapped to the mitochondrial genome. Dependent on the specific metric considered, either low or high values serve as indicators of poor-quality cells. For instance, low values of library size may suggest poor sequencing or the presence of dead cells, while an abnormally high may come from doublets. When concerning the fraction of mitochondrial genes, only high values are problematic as they may indicate apoptotic cells or cells with broken membranes during sequencing. Since if cells are broken, cytoplasmic mRNAs leak out, and only larger mitochondrial mRNAs are sequenced which are less likely to escape through tears in the cell membrane. Low quality cells are filtered out by thresholding either fixed or adaptive (data-driven) (**Figure 24**) [95].

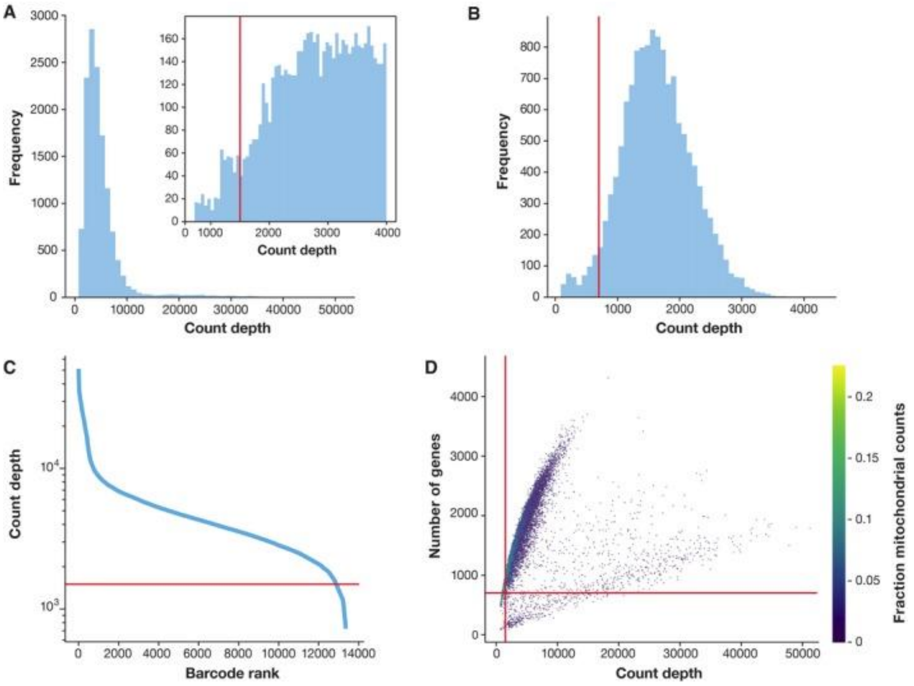


Figure 24. Examples of QC plots with fixed thresholds [93]. Thresholds (red lines) are determined based on prior knowledge about the biological system under investigation. (A) Histograms of total number of counts per cell. The smaller histogram is zoomed-in on values below 4,000. (B) Histogram of the number of genes detected per cell. Cells are filtered out at 700 genes. (C) Count depth distribution from high to low count depths. It shows an "elbow" where count depths start to decrease rapidly around 1,500 counts. (D) Number of genes versus the count depth colored by the fraction of mitochondrial reads. Mitochondrial read fractions are only high in particularly low count cells with few detected genes. These cells are filtered out.

Normalization

Following quality control, the expression matrix still not accurately reflect the true biological gene expression levels due to technical noise. To address this issue and make the raw read counts informative and comparable across cells, a normalization strategy must be applied. The primary objective of normalization is to remove systematic technical biases from the data.

Usually, between-sample normalization is performed, to account for variations in sequencing depth between cells [64]. Different assumptions are taken to produce normalized values, such as the total amount of mRNA per cell or the level of 'symmetry' in differential expression [101]. Computational normalization strategies are broadly categorized into two groups:

- Global scaling approach: this approach, inherited from bulk RNAseq, includes methods like Reads per Million mapped reads (RPM), Transcripts Per Kilobase Million (TPM) [102], DESeq [103] and Trimmed Mean of M-values (TMM) [104]. The idea behind this approach is to calculate a single scaling factor (size factor) for each cell, representing the estimated relative technical bias in that cell. The counts for each cell are then divided by its corresponding scaling factor to obtain normalized values. However, directly applying this strategy to scRNAseq data poses challenges due to the dominance of low and zero counts, making the estimation of scaling factors less stable [64].

- Tailored approaches for scRNAseq data: these methods incorporate various ideas to address scRNAseq-specific biases. Examples include sctransform [61], SCnorm [62] and scran [63].

To overcome the impact of problematic zero counts, a deconvolution strategy has been proposed (implemented in the scran method). This involves summing counts from multiple cells (pooling) and normalizing them against an average reference (**Figure 25**) [63].

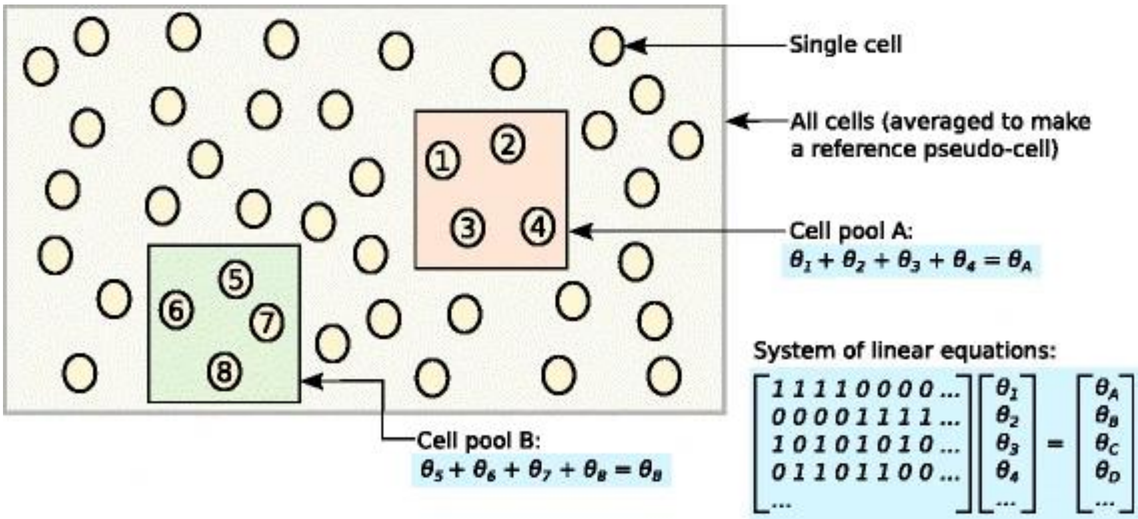

Single cell
All cells (averaged to make a reference pseudo-cell)
Cell pool A:
$\theta_1 + \theta_2 + \theta_3 + \theta_4 = \theta_A$
1
2
3
4
5
6
7
8
Cell pool B:
$\theta_5 + \theta_6 + \theta_7 + \theta_8 = \theta_B$
System of linear equations:
$\begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0 \ldots \\ 0\,0\,0\,0\,1\,1\,1\,1 \ldots \\ 1\,0\,1\,0\,1\,0\,1\,0 \ldots \\ 0\,1\,1\,0\,1\,1\,0\,0 \ldots \\ \ldots \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \ldots \end{bmatrix} = \begin{bmatrix} \theta_A \\ \theta_B \\ \theta_C \\ \theta_D \\ \ldots \end{bmatrix}$

***Figure 25**. Deconvolution strategy to normalize scRNAseq data [63]. All cells in the data set are averaged to make a reference pseudo-cell. Expression values for cells in pool A are summed together and normalized against the reference to yield a pool-based size factor θ A. This is equal to the sum of the cell-based factors θ j for cells j=1–4 and can be used to formulate a linear equation. (For simplicity, the t j term is assumed to be unity here.) Repeating this for multiple pools (e.g., pool B) leads to the construction of a linear system that can be solved to esti-mate θ j for each cell j*

Bacher et al [62] and Hafemeister et Satija [61] have emphasized that treating all genes equally is unjustified due to the relationship between a gene expression (count) and cellular sequencing depth. This phenomenon is known as the count-depth relationship. A single global scaling factor is unable to accommodate this relationship adequately. As a result, they propose different solutions to address this issue. The SCnorm method developed by Bacher et al. adopts a scaling approach. It groups genes with similar count-depth relationships and applies separate scaling factors to each group. Such approach is known as gene group-specific normalization. However, SCnorm does not account for zero values in the data.

In contrast, the sctransform method introduced by Hafemeister et Satija transforms the data using Pearson residuals derived from a regularized negative binomial regression. Nevertheless, there have been concerns raised about the performance of sctransform, with some studies suggesting that it may lead to a high proportion of false discoveries in differential gene expression analysis [105] [106].

Normalized values are often log-transformed to reduce skewness in their distribution, and a pseudo-count of 1 is added to each normalized value to avoid undefined values at zero.

Feature selection

A typical scRNAseq dataset consists of tens of thousands of genes, also referred to as features. However, only a small fraction of these genes is associated with the cell's response to the biological factor of interest, while the majority contain random noise. The presence of noisy features can hinder downstream biological analysis such as clustering. Therefore, it is essential to remove these noise-driven genes while preserving biologically relevant information. This process is referred to as a feature selection or highly variable genes (HVGs) selection. The underlying assumption is that genuine biological differences will be manifested by higher levels of variation in the genes of interest, in contrast to other genes that are primarily affected by technical noise. However, because of heteroscedasticity (a positive relationship between the mean expression of a gene and the variance), variance cannot be used as a direct indicator of HVGs. Therefore, different methods for HVG selection employ various measures of variability.

basic approach is based on modelling of the mean-variance relationship. To address heteroscedasticity, the variance of log-normalized expression values is modelled. By fitting a mean-variance trend, the variance is decomposed into technical components captured by the trend and biological components represented by the residuals from the trend (**Figure 26**) [107]. However, large biological components can also be attributed to “housekeeping” genes, which are considered not relevant for characterizing cellular heterogeneity.

Various flavors of the above strategy are utilized. For instance, Brennecke [108] uses the squared coefficient of variation (CV^2) instead of variance, while M3Drop [67] identifies HVGs based on genes with a dropout rate exceeding that of other genes with the same mean expression. However, M3Drop may not detect highly expressed genes since these may have no dropouts, even when they are differentially expressed across cell populations.

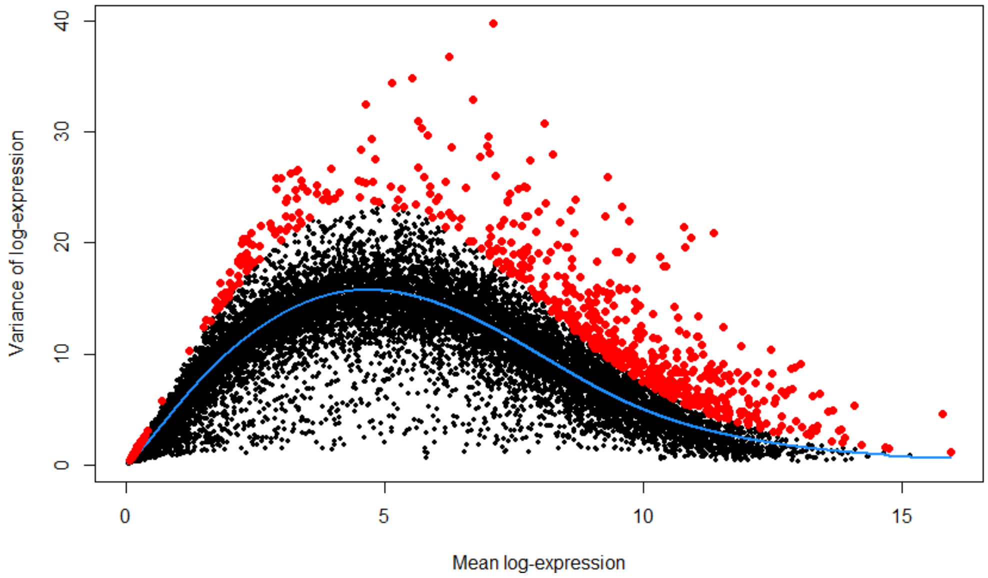


Figure 26. Mean-variance modelling for feature selection in scRNAseq [107]. Each point represents a gene while the blue line represents the trend (technical noise) fitted to all genes. Red points represent HVGs.

A different approach to feature selection is based on gene correlations [109]. Instead of testing genes individually, the method called DUBStepR examines relationships between gene expression in a stepwise manner. The underlying assumption is that differentially expressed genes specific to the same cell types should exhibit high correlations, while those specific to distinct cell types should have low correlations. For non-differentially expressed genes, the correlations should be weak. Following this assumption, an initial set of features is selected, and stepwise regression is performed by iteratively removing the gene that explains the most variance in the residual from the previous step. One limitation of this method is that it assumes technical noise to be random and independent for each cell, thereby not leading to gene correlations. However, this assumption is violated in the presence of batch effects.

Dimensionality reduction

Depending on the specific downstream biological analysis, the dimensionality of the single-cell expression matrix can be further reduced through a process known as feature extraction or 'dimensionality reduction' The main distinction between feature selection and feature extraction lies in their approach: while feature selection preserves the original biologically relevant features unchanged (**Figure 27B**), feature extraction combines the original feature space into a new, smaller set (**Figure 27A**). This transformed representation, intended to capture the underlying structure, is referred to as a manifold or low-dimensional representation. (**Figure 27C**).

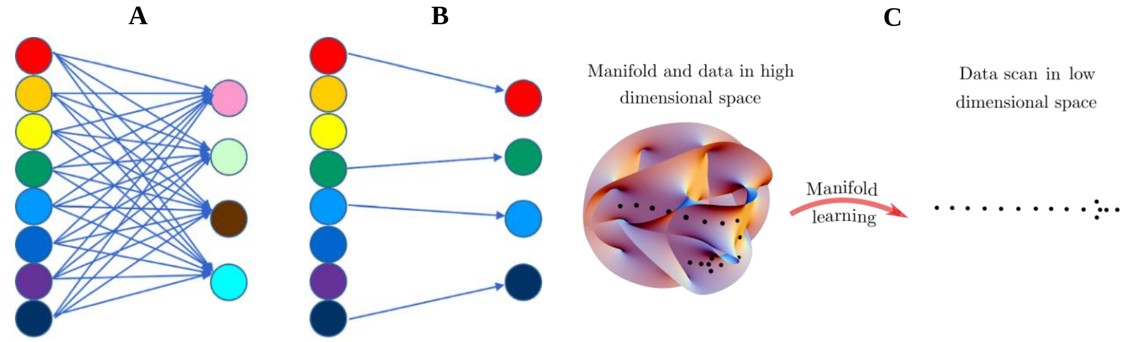

A
B
C
Manifold and data in high dimensional space
Data scan in low dimensional space
Manifold learning

***Figure 27**. The idea of dimensionality reduction. A – in feature extraction, each dimension is a complex combination of many genes, whereas in feature selection, original features considered relevant are chosen from the original set, B – schematic of manifold learning. This process utilizes principles of data compression and facilitates data visualization.*

Some dimensionality reduction methods, such as principal component analysis (PCA) [110], are generic, while others, like ZIFA [69] and ZINB-WaVE [111] are specifically tailored for scRNAseq data. The most general division of dimensionality reduction algorithms is based on whether they perform manifold learning through linear or nonlinear combinations of features. A commonly used linear method is PCA, which identifies principal components (PCs) that explain the maximum variability in the data through orthogonal transformation. However, the first few components often correlate with the number of detected genes rather than the biological signal of interest [68, 111]. Although PCA is highly interpretable and widely used as a preprocessing step for clustering, it assumes approximately normally distributed data (an unrealistic assumption for scRNAseq data) and performs poorly on sparse matrices where distant points can become nearest neighbors.

Generally non-linear algorithms are better suited for scRNAseq data. Two leading methods are t-distributed stochastic neighbor embedding (t-SNE) [112] and Uniform Manifold Approximation and Projection (UMAP) [113]. t-SNE focuses on capturing local neighborhoods by grouping neighboring data points together, often at the cost of preserving global data structure. UMAP, on the other hand, attempts to strike a balance between local and global structure. t-SNE is computationally expensive (quadratic time and space complexity with respect to the number of data points), has a stochastic nature due to random initialization, and can only embed data points into a maximum of three dimensions. As a result, it is recommended by the authors for visualization purposes rather than general dimensionality reduction.

t-SNE focus on capturing local similarity that is to group neighboring data points together but at the cost of global structure whereas UMAP tries to achieve a trade-off between local and global structure. t-SNE is computationally expensive (quadratic time and space complexity in the number of data points), has stochastic nature (random initialization) and embeds data points onto maximum 3 dimensions, hence it is recommended by the authors to use only for visualization purposes (not for general dimensionality reduction).

UMAP overcomes some of the limitations of t-SNE. It is less changing between runs, as it does not rely on fully random initialization. Additionally, UMAP is faster and less computationally expensive due to the application of stochastic gradient descent. Unlike t-SNE, UMAP can embed data points onto more than three dimensions. However, both algorithms share a lack of strong interpretability, as distances between clusters may not hold specific meaning. Moreover, tuning the hyperparameters of t-SNE and UMAP is necessary for optimal performance [114].

Clustering

The final step in the general analysis stage is clustering, which is an unsupervised learning technique used to group cells with similar expression profiles into clusters. Clustering plays a crucial role as many subsequent biological analyses rely on its results. Once cells are grouped into clusters, they can be further annotated. However cell annotation will not be discussed here.

The concept of similarity forms the foundation of any clustering method. However, in the original high-dimensional space of scRNAseq data, the distances between cells become similar, which is known as the 'curse of dimensionality'. Therefore, similarity is usually measured in dimensionality-reduced representations of the data.

Two broad classes of similarity metrics can be distinguished:

- Distance-based: these metrics consider objects with the lowest values as the most similar, such as Euclidean, Manhattan, or Hamming distance. It should be noted that distance-based metrics for continuous data are sensitive to scaling.
- Correlation-based: these metrics consider objects with the highest values as the most similar, such as Pearson's correlation, Spearman's correlation, or cosine similarity. Importantly, correlation-based metrics are invariant to scaling.

The choice of a specific similarity metric can significantly impact the outcomes of clustering analysis, with Pearson's correlation often demonstrating the highest overall performance [115].

Clustering approaches to scRNAseq are categorized into the following groups:

- partition-based
- hierarchical-based,
- graph-based

Often, a particular algorithm for dimensionality reduction, combines two or more approaches, for example pcaReduce which is the combination of PCA, k-means and hierarchical-based clustering [116].

K-means clustering is an iterative procedure that aims to partition a dataset into k distinct, non-overlapping clusters, where each data point is assigned to only one group (**Figure 28**) [117].

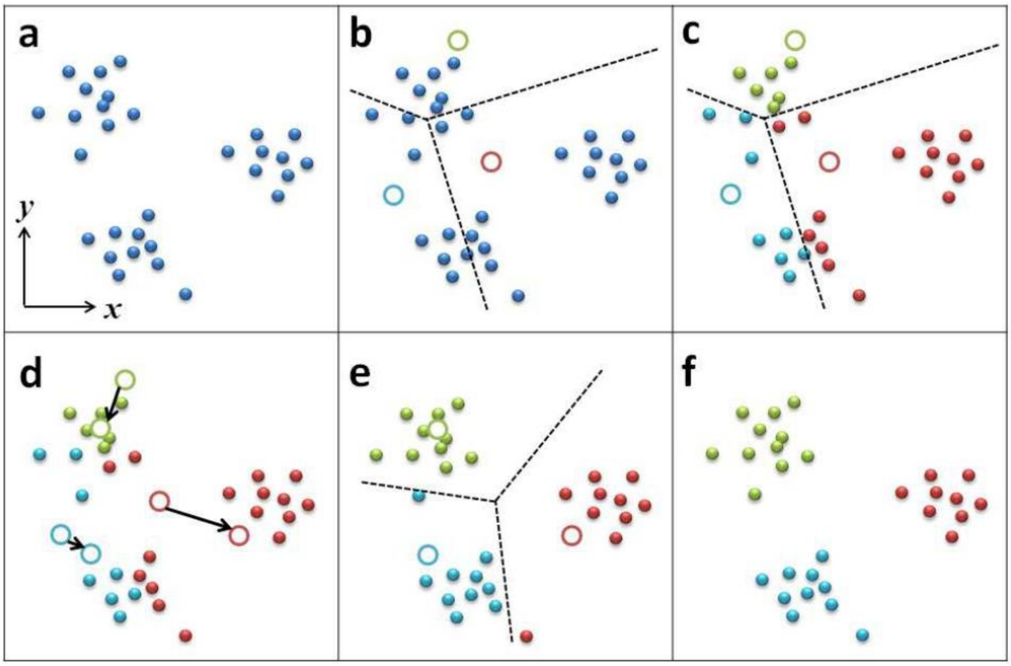


Figure 28. *A schematic illustration of the k-means algorithm for two-dimensional data clustering [117]. (a) The data points (solid blue circles) to be clustered in a 2D feature space. (b) The algorithm is initialized by finding k (pre-defined) random representatives of clusters called centroids (aqua, green, and red hollow circles) (c) Each cell is assigned to the cluster with the closest centroid based on Euclidean distance, which is done by minimizing the within-cluster sum of squares (d) based on the assignments the mean of each cluster is calculated and based on which the position of centroid is updated (movements shown by the black arrows), (e) Each cell is assigned to the cluster with the new closest centroid (f) final assignments*

The k-means concept is employed in several tools designed for scRNAseq data clustering. For instance, the SC3 method [118] combines results from multiple runs of k-means clustering with different parameter combinations to create a consensus matrix, which enhances the accuracy of cell type identification but at the cost of increased computational requirements.

One drawback of k-means clustering is its preference for spherical clusters with equal radii, which can lead to rare cell types being merged with larger groups. To address this issue, RaceID [119] combines outlier detection methods with k-means to improve clustering accuracy. Another limitation of k-means clustering is its equal weighting of both biologically relevant and irrelevant (noisy) features, which can blur the cluster structure [120]. Although applying a dedicated dimensionality reduction algorithm can mitigate this problem, it may also result in the loss of valuable biological information and pose challenges for interpretation [121]. To overcome this, an enhanced version of k-means called sparse k-means was proposed, which adaptively selects and reweighs a subset of features during the clustering process [122-124].

Hierarchical clustering is another generic clustering procedure, commonly used for aggregating data. This method constructs a hierarchy of clusters using either an agglomerative or divisive approach. In addition to the similarity measure between objects, a linkage measure between clusters (e.g., single, complete, average, or Ward linkage) must be specified. The results of hierarchical clustering are typically visualized in the form of a tree called dendrogram, and clusters are obtained by cutting the tree at a desired level. While hierarchical clustering produces reproducible results, it is not suitable for large datasets due to its quadratic time complexity. It is adapted in methods such as CIDR [70] or pcaReduce [116].

To address the scalability issues graph-based methods have been employed for scRNAseq clustering. A graph is a structure where each node represents a cell and edges represent similarities between cells. Following graph construction, community detection is performed to identify groups of nodes that are more connected within the same community than to nodes in different communities. Graph-based clustering does not assume specific cluster shapes or cell distributions within each cluster. The resolution of clustering can be controlled by specifying the minimum number of nearest neighbors each cell should be connected to. Additional parameters include the weighting criteria for edges and the choice of community detection algorithm (e.g., walktrap, Louvain, fast_greedy). PhenoGraph [125] and SNN-Cliq [126] are examples of graph-based clustering methods. Figure 35 illustrates the differences in clustering results obtained by different approaches.

Published evaluations of different clustering methods reveal the subjective nature of clustering as a task. The performance of clustering tools varies in terms of stability, agreement with true partitions, running times, and overall effectiveness, which can be influenced by preprocessing steps and dataset complexity [127-129]. Estimating the appropriate number of clusters is also a challenging aspect, as there is always the possibility of performing more splits. One useful approach to address this challenge is to examine the number of significant differentially expressed (DE) genes obtained from each subsequent split. Certain tools, such as SNN-Cliq, provide automated estimations for the number of clusters, but these estimations may not always have biological relevance. At the core of each clustering procedure lies the assumption that discrete clusters exist within the data [130]. As a result, any algorithm will identify some form of grouping, whether it is biologically meaningful or not.

III. Methods

III.1 Experimental design

The datasets used in this PhD project were derived from two related scRNAseq experiments conducted to explore the impact of navitoclax treatment on the transcriptome of a triple-negative breast cancer cell line. The objective was to gain a deeper understanding of the mechanisms underlying the development of drug resistance [28, 131]. Both experiments utilized the MDA-MB-231 cancer cell line, and two biological replicates, labelled as A and B, were included. The cells were subjected to a 10 µM concentration of navitoclax and harvested at three specific time points:

1) before the treatment (baseline; T1),

2) after treatment (T2),

3) after recovery from the treatment (T3)

In both experiments, the initial step involved trypsinizing the cells and preparing a single-cell suspension with a concentration of 1,000 cells/µl and viability above 90%. The single-cell libraries were prepared using the droplet-based platform from 10X Genomics. The following reagents were used: Chromium Single Cell 3′ Library and Gel Bead Kit V2 (PN-120237), Chromium Single Cell A Chip Kit (PN-120236), and Chromium i7 Multiplex Kit (PN-120262). For sequencing, the same Illumina HiSeq 4,000 sequencer was utilized.

In the study conducted by [28], a total of 6,000 cells per sample were used, with two samples multiplexed on one lane. This generated 25,000 reads per cell. The second study sequenced 1,500 cells per sample in a single lane, resulting in 200,000 reads per cell.

Both experiments shared the same biological properties, including the cell line, drug treatment, and harvesting time. The only difference between them was in the technical study design, as illustrated in **Figure 29**.

The first experiment [28] was part of a larger study. However, upon analysis, it was discovered that the experiment exhibited strong batch effects resulting from variations in the experimental processing of the biological groups corresponding to the time of harvesting. In this study, cells collected at different time points were processed on separate chips and on various days. For this PhD study, only the repetition A from this experiment was considered.

The second experiment, referred to as a balanced study [131], was designed to minimize technical variation. In this design, cells collected at different time points were split and processed on the same chip, all on the same day. This approach aimed to ensure that any observed differences were primarily due to the effects of the drug treatment and not influenced by technical factors. This study serves as a reference.

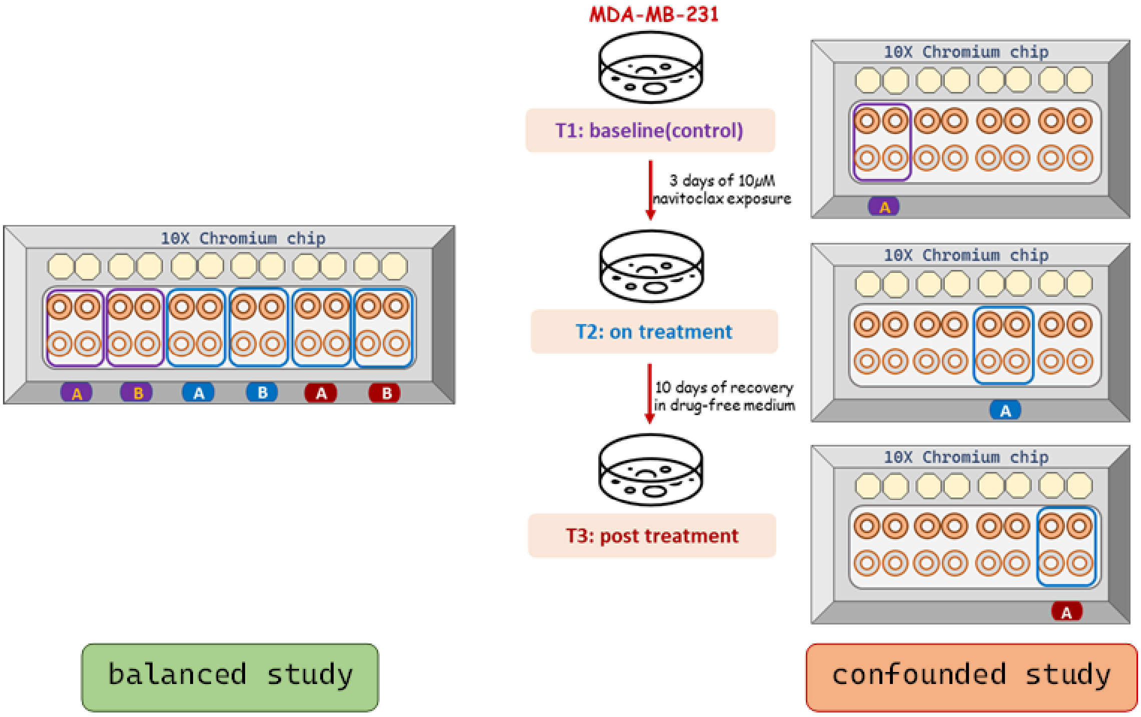


Figure 29. Experimental design.

III.2 Preprocessing and data cleaning

The quality of raw reads in the form of FASTQ files was evaluated using FastQC software from Babraham Institute, UK. Alignment and transcript quantification were performed by executing ‘`cellranger count`’ function from Cell Ranger software (v.6.1.1 from 10x Genomics). The obtained expression matrix was imported into the R environment and then subsetted to include only the protein coding genes.

A second round of quality control (QC) was conducted to filter out dead or broken cells, as well as any "cells" that may have resulted from technical issues during library preparation, such as doublets or empty droplets. The following metrics were evaluated for each timepoint separately: library size, number of detected genes per cell, and the proportion of reads mapped to the mitochondrial genome. Adaptive thresholds were applied to these QC metrics based on the median absolute deviation (MAD) from the median value of each metric. If a metric value deviated more than 3 MADs from the median in the problematic direction, it was considered an outlier.

Following QC and discarding poor-quality cells, UMI counts across cells were normalized using a deconvolution approach implemented in the scran R package. The normalized counts were log-transformed, and a pseudo-count of 1 was added to each item.

Feature selection for batch effect correction was performed using the ‘vst’ method from Seurat [61], which is a variance stabilization technique. This method fits a line to the relationship between log (variance) and log (mean) using local polynomial regression. The feature values are then standardized using the observed mean and expected variance obtained from the fitted line. This process resulted in the identification of 3,620 highly variable genes. This set of highly variable genes (HVGs) was exclusively used as input for the batch correction algorithms.

III.3 Batch-effect correction methods

Six algorithms that met the criteria of producing a corrected expression matrix and working in an unsupervised manner were benchmarked. For simplicity, the evaluation of batch effect correction methods focused on two time points (T1 and T2) from both datasets, balanced/confounded.

ComBat-seq

ComBat-seq [85] requires two input parameters: an untransformed count matrix and a vector that describes the cell annotation into batches. It also allows for the specification of biological covariates, which will be preserved in the corrected data. In our study, the batch separation vector was associated with the repetition, while the time point was associated with the biological variable. ComBat-seq uses a negative binomial regression model to estimate batch effects. The computed batch-effect estimators are then used to calculate “batch-free” distributions, representing the expected distributions if there were no batch effects in the data based on the model. ComBat-seq applies quantile normalization to ensure that the empirical and batch-free distributions have identical statistical properties.

Limma

Limma [86] takes normalized and log-transformed counts as an input and works by incorporating the batch information into the linear model to account for the batch effects. This model is aimed at capturing the variation attributed to both the biological factors and the batch effects. Then empirical Bayes methods are applied to shrink the estimated coefficients towards the overall mean. The batch effects are subsequently subtracted from the original data, resulting in the batch-corrected expression matrix.

Mutual nearest neighbor (MNN)

MNN method [89] requires that a subset of the population is shared between batches. The minimum size of this shared subpopulation, denoted as 'k', is defined by the user. MNN pairs are formed by identifying the most similar cells of the same type/state across batches. These pairs consist of cells with mutually similar expression profiles, making any differences between them likely to be driven by batch effects. In the MNN correction method, two batches are considered at a time. Based on this pair of batches, a correction vector is estimated as the difference in expression values between cells in an MNN pair. This means that one of the batches always serves as a reference batch, which subsequent batch is merged to. This integrated dataset serves as a new reference to iteratively integrate more datasets.

MNN correction was performed with using ‘`mnnCorrect`’ function from ‘batchelor’ R package. The function was run with two setups: one with all genes and another with highly variable genes (HVGs). For both cases, normalized and log-transformed expression values were used as input.

The ‘merge.order’ argument was specified to ensure that both repetitions from a given time point were merged first and then combined. The merging order followed this sequence: first T1A + T1B and T2A + T2B. The results of these summations were then added together.

To obtain corrected values on the log scale, similar to the input data, the 'cos.norm.out' parameter was set to FALSE. The remaining parameters were kept at their default values.

scMerge

The framework of scMerge involves the following steps: (i) identification of stably expressed genes (SEGs) that act as "negative control genes" across batches, (ii) k-means clustering across batches based on the highly variable genes (HVGs), (iii) identification of pairs of mutual nearest clusters (MNCs) across batches using Pearson correlation as the dissimilarity metric; cells belonging to a pair of MNCs are considered to be of the same type and serve as pseudo replicates, (iv) factor analysis using the SEGs and pseudo replicates as inputs to generate a single merged dataset.

scMerge correction was performed using the 'scMerge' function from the R package of the same name. Three setups of the 'kmeansK' parameter were used: (5, 5, 5, 5), (4, 4, 4, 4), and (4, 4, 3, 3).

Seurat v4

Seurat v4 [132] includes two approaches for MNN or anchors matching across batches: Canonical Correlation Analysis (CCA) and reciprocal Principal Component Analysis (rPCA).

In both approaches, the search for anchors is conducted in a shared and reduced subspace obtained through CCA (linear combinations of genes with the highest correlation between batches) or rPCA (maximizing the variation between batches). The correction vector is computed by comparing the expression profiles of cells within each anchor. The order of batch integration is determined using hierarchical clustering based on the distance between the datasets.

Scanorama

The idea of Scanorama [90] draws inspiration from the image stitching technique used in computer vision, where overlapping images are merged into a larger panorama. Scanorama extends the concept of MNN matching to identify similar cells across multiple batches [88]. Scanorama performs the search in a low-dimensional subspace obtained through randomized singular value decomposition (SVD), encompassing all batches simultaneously. The priority of batch merging is determined by the percentage of matching cells within each batch. Two setups were evaluated: one using all genes as input and another using the top 2,000 highly variable genes identified based on data dispersion.

III.4 Visualizations

Clustering visualizations were performed using the Uniform Manifold Approximation and Projection (UMAP) [133]. An R implementation of the UMAP through ‘uwot’ package was used. Crucial hyperparameters affecting the visualization were set as follows; the number of neighbors: `n_neighbors` = 15, the minimum distance between embedded points: `min_dist` = 0.01. The input for UMAP consisted of a normalized and log(x+1)-transformed gene expression matrix. Expression values for each gene were scaled to the range of [0,1] across cells. The similarity between cells was calculated using the Euclidean distance.

Sankey diagrams were generated using a tool called SankeyMATIC (https://sankeymatic.com/).

III.5 Proposed pipeline

The framework utilizes the concept called Divisive Intelligent K-Means (DiviK) [134], which is based on iterative clustering with a 2-step feature space optimization. However, this framework was adjusted and originally enhanced to address scRNAseq data. The original improvement involved combining DiviK with functional analysis of gene pathways and a cluster linkage procedure (**Figure 30**). This approach enables the analysis of completely confounded scRNAseq datasets without the need for potentially harmful batch correction.

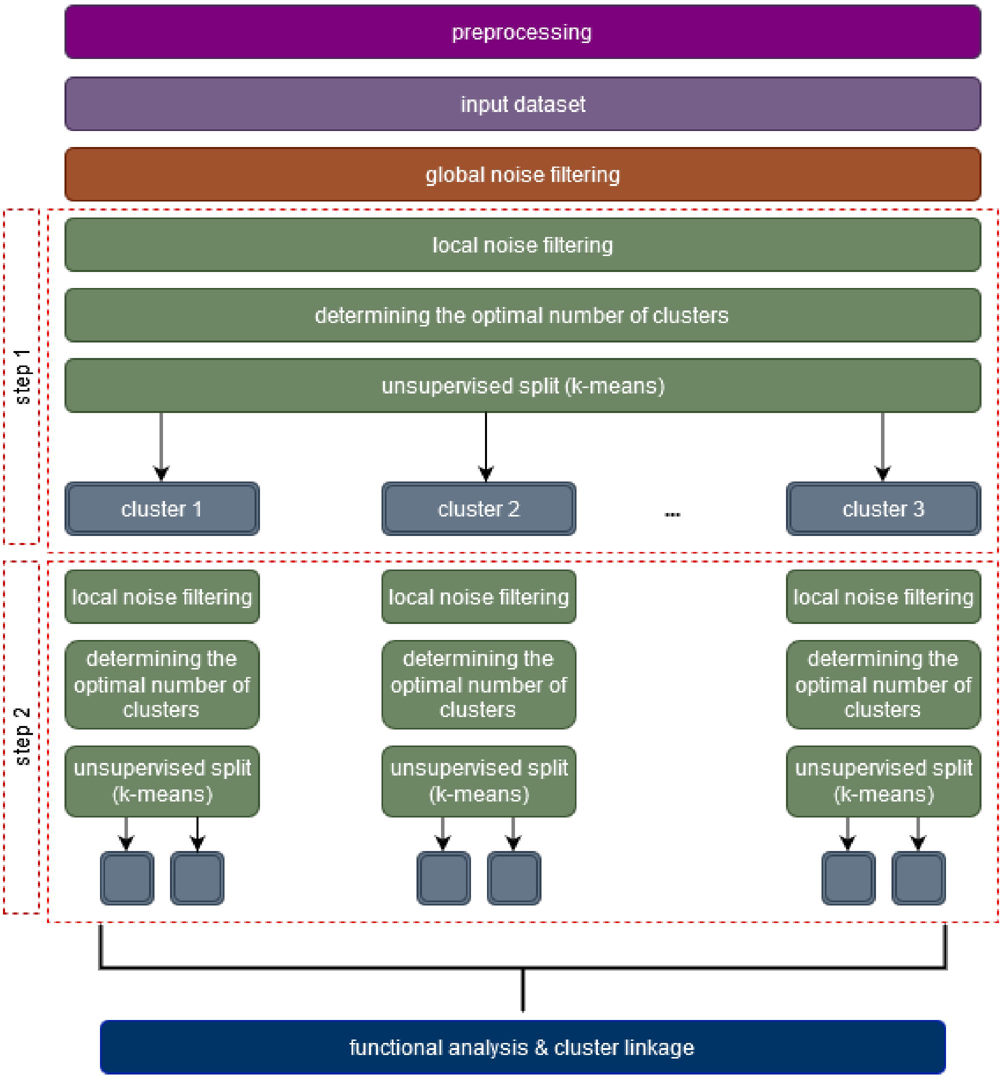

preprocessing
input dataset
global noise filtering
step 1
local noise filtering
determining the optimal number of clusters
unsupervised split (k-means)
cluster 1
cluster 2
...
cluster 3
step 2
local noise filtering
determining the optimal number of clusters
unsupervised split (k-means)
local noise filtering
determining the optimal number of clusters
unsupervised split (k-means)
local noise filtering
determining the optimal number of clusters
unsupervised split (k-means)
functional analysis & cluster linkage

Figure 30*. The proposed framework for analysis of confounded datasets. The first step involves a filtering procedure that addresses the prevalent issue of zero measurements in scRNAseq data. This filtering is performed only once and globally, meaning it is applied to the entire feature space. The goal of global filtering is to remove significant noise from the data. In the second step of feature selection, a filtering strategy is applied locally to each cluster obtained at every clustering iteration. Clusters discovered within batches are subsequently subjected to independent functional analysis of gene sets. Following functional analysis, clusters from corresponding timepoints in the reference and confounded datasets are linked based on the similarity of their functional profiles.*

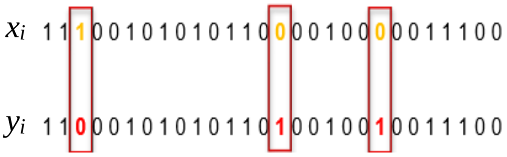
III.5.1 Global noise filtering

Genes with high frequencies of zero measurements are the main source of global noise in scRNAseq data. To avoid applying fixed thresholds on dropout rates, data-driven filtration was utilized using hierarchical clustering (HC). The HC was performed on binarized gene expression. Binarization (non-zero counts were changed to ones) was performed on the raw UMI counts after discarding poor-quality cells. Subsequently, a Hamming distance matrix was computed between every pair of genes, which were now represented as binary strings, using the formula below:

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$


x_i 1 1 1 0 0 1 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0
y_i 1 1 0 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0 0 1 0 0 1 1 1 0 0

x_i, y_i – a pair of binary strings, k – strings length (must be the same), **D_H** – Hamming distance
The distance between two binary strings (a pair of genes) is the number of positions at which the corresponding bits are different. In the above example: **D_H (x_i, y_i) = 3**

The R function '`parallelDist`' was utilized to compute the Hamming distance. This function performs calculations in parallel using multiple threads. Complete linkage was chosen as an agglomeration method which is based on maximum distance that is, merging is performed between clusters with the smallest maximum distance between their elements (farthest neighbors). Function '`hclust`' from fastcluster R package was utilized for HC. Global filtration was performed only once and applied dataset wide (to the entire feature space). The resulting feature space, after global filtration, is referred to as the 'reduced domain.' In contrast, the term 'full domain' is used to describe the original feature space before any filtration occurred.

III.5.2 Local noise filtering

Local noise filtering, also referred to as local space optimization or feature space engineering was performed by decomposing gene variances into a mixture of Gaussian components using Gaussian Mixture Models (GMM) [4]. The optimal number of components was determined using the Bayesian Information Criterion (BIC) within a range from 1 to 10.

Each gene was assigned to a specific Gaussian component using the maximum a posteriori (MAP) rule. The components were then ordered based on their location parameter, which represents the mean of the Gaussian component. Genes corresponding to components located at the rightmost side of the signal scale were considered highly variable genes. The threshold was determined by the intersection between the two right most components. Conversely, the components located at the left-hand side were deemed non-informative for further analysis.

Each subsequent GMM filtering started from the reduced domain, ensuring that the original information was still available regardless of the current depth of the analysis. This step was performed by executing '`normalmixEM`' function from mixtools R package.

III.5.3 Unsupervised splitting

Unsupervised split was performed by employing two variants of k-means clustering: classic k-means and the more modern sparse k-means. The use of sparse k-means aimed to validate the GMM filtering approach, as this variant incorporates a feature selection procedure. In other words, it internally assesses the importance of each gene in the clustering process by assigning higher weights to more significant genes.

The '`KMeansSparseCluster`' function from *sparcl* R package [122] was utilized for sparse clustering and '`kmeans`' function from *base* R package for classic kmeans. In both cases, the maximum number of iterations was set to 10. To mitigate the impact of highly expressed genes on the clustering results, the log-normalize gene expression values were scaled to the interval [0,1] prior to clustering.

III.5.4 Determining the optimal number of clusters

The optimal number of clusters was determined using the Calinski-Harabasz index, also known as the variance ratio criterion (VRC) [56]. Clustering was performed with various values of k, ranging from 2 to 10 clusters. The optimal number of clusters was determined by selecting the clustering with the highest value of the Calinski-Harabasz index. This metric was calculated using the following formula:

$$VRC_k = \frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)}$$

$$SS_B = \sum_{i=1}^{k} n_i \|m_i - m\|^2 \qquad SS_W = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - m_i\|^2$$

where:

SS_B is the overall between-cluster variance, **SS_W** - the overall within-cluster variance, **k** - the number of clusters, **N** - the number of observations, **n_i** - the number of observations in cluster *i*, **m_i** - the centroid of cluster *i*, **m** - the overall mean of the sample data, and **$\|mi-m\|$** is the L2 norm (Euclidean distance) between the two vectors.

III.5.5 Functional analysis and cluster linkage

Functional analysis on discovered clusters was conducted through gene set variation analysis (GSVA) [135]. GSVA is an unsupervised technique that takes a log-normalized gene expression data matrix as input. It calculates the pathway enrichment score for each cell by comparing the empirical cumulative distribution functions (CDFs) of gene expression ranks within and outside the gene set. Enriched pathways, indicating up-regulation, are assigned positive values for the enrichment score, while down-regulated pathways receive negative values. Pathways from the Kyoto Encyclopedia of Genes and Genomes (KEGG) were utilized. This collection consists of 186 pathways. The GSVA R package was employed, specifically using the '`gsva`' function.

Prior to the analysis, a filtration step was applied, involving the removal of the following:

- Genes from the input expression data matrix with constant expression.
- Genes from the input gene sets that do not have a corresponding gene in the input gene expression data matrix.
- Gene sets that do not meet the user-specified minimum and maximum size requirements. The minimum and maximum gene set size was set to 15 and 500, respectively.

To find cluster specific pathways, Cliff's delta effect size statistics was proposed. This metric quantifies the extent to which values in one group are larger (dominate) than the values in a second group. However, these groups were determined according to one-versus-others scenario. This means that one group consisted of vector of enrichment scores for a specific cluster, while the second group consisted of a vector of enrichment scores for all the other clusters. Calculations were performed across all discovered clusters. Cliff's delta effect size statistics is defined through delta function (δ), as follows:

$$\delta(i,j) = \begin{cases} +1, & x_i > y_j \\ -1, & x_i < y_j \\ 0, & x_i = y_j \end{cases}$$

$$\delta = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \delta(i,j)$$

where:

Two groups are defined: X = ($x_1, \ldots, x_m$) and Y = ($y_1, \ldots, y_n$),

m,n – size of group X and Y respectively

Cliff's delta ranges from -1 to 1 and does not rely on assumptions regarding the shape or spread of the two groups being compared. The interpretation of this metric does not follow strict rules and should each time be adapted to the given experimental setup.

Then scoring function was constructed with using previously calculated metrics to enable cluster linkage. Two variants of scoring function were utilized:

- based on similarity score and calculated as follows:

$$similarity\ score = \sum_{All\ pathways} \delta_{C1} * \delta_{C2}$$

$\boldsymbol{\delta_{C1}, \delta_{C2}}$ - values of Cliff's delta effect size statistics for each pathway in cluster C1 and C2 respectively

- based on Pearson correlation coefficient:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where:

- n: numer of pathways
- x_i, y_i: values of Cliff's de
- lta effect size statistics for each pathway in cluster C1 and C2 respectively

IV. RESULTS

IV.1 Evaluation of batch effect correction methods

Both datasets (balanced and confounded) were visualized at the cell level through UMAP plots (**Figure 31**). In the balanced dataset, cells from both repetitions group according to the biological variable of interest (timepoint). However, in the confounded dataset, each technical replicate forms its own cluster. This indicates that the dataset is completely confounded, with batch effects overpowering the biological variable of interest.

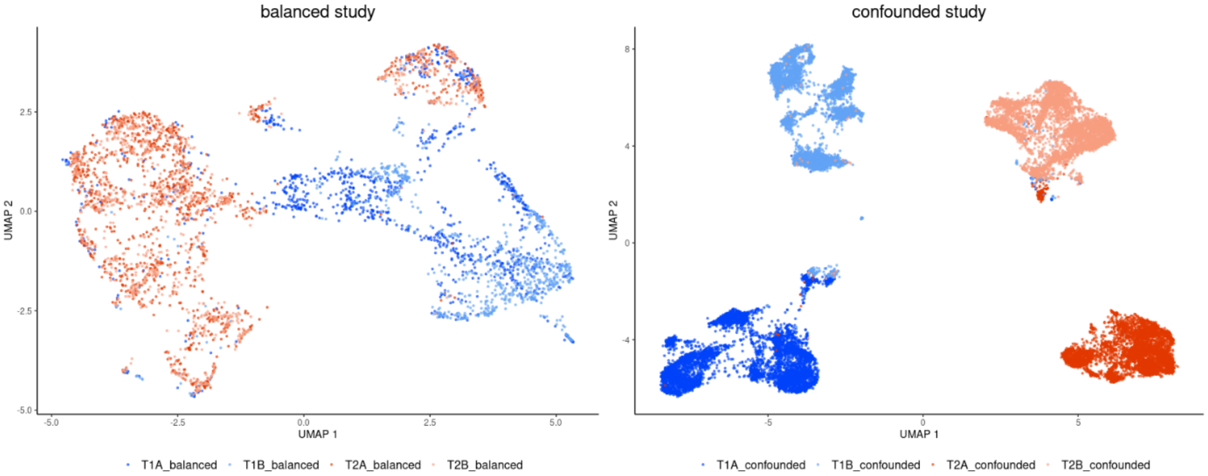


Figure 31. *UMAP plots of balanced and confounded study.*

Since reliable analysis of such a completely confounded dataset is not possible, six batch-effect correction tools were applied to address this issue (**Figure 32**). ComBat-seq resulted in two distinct clusters corresponding to different timepoints, with the cells from technical repetitions well intermingled. Limma also showed strong segregation based on the biological variable, but the repetitions appeared to cluster separately rather than intermingling. The MNN algorithm improved the separation by timepoint in both scenarios, considering all genes and only the top 3,620 highly variable genes (HVGs). However, cells within T1 were likely to form subgroups. scMerge also improved the separation by timepoint compared to no correction across all tested setups of kmeansK parameter. However, the best performance was achieved at kmeansK = (4,4,3,3) where cells grouped by timepoint as well were well intermingled within technical replicates. For other setups of kmeansK, the technical replicates from T2 (A and B) clustered separately. Seurat achieved the worst result by mixing all cells together in both tested scenarios, thus it was discarded from further analysis. Scanorama achieved little improvement, regardless of whether all genes or only HVGs were used.

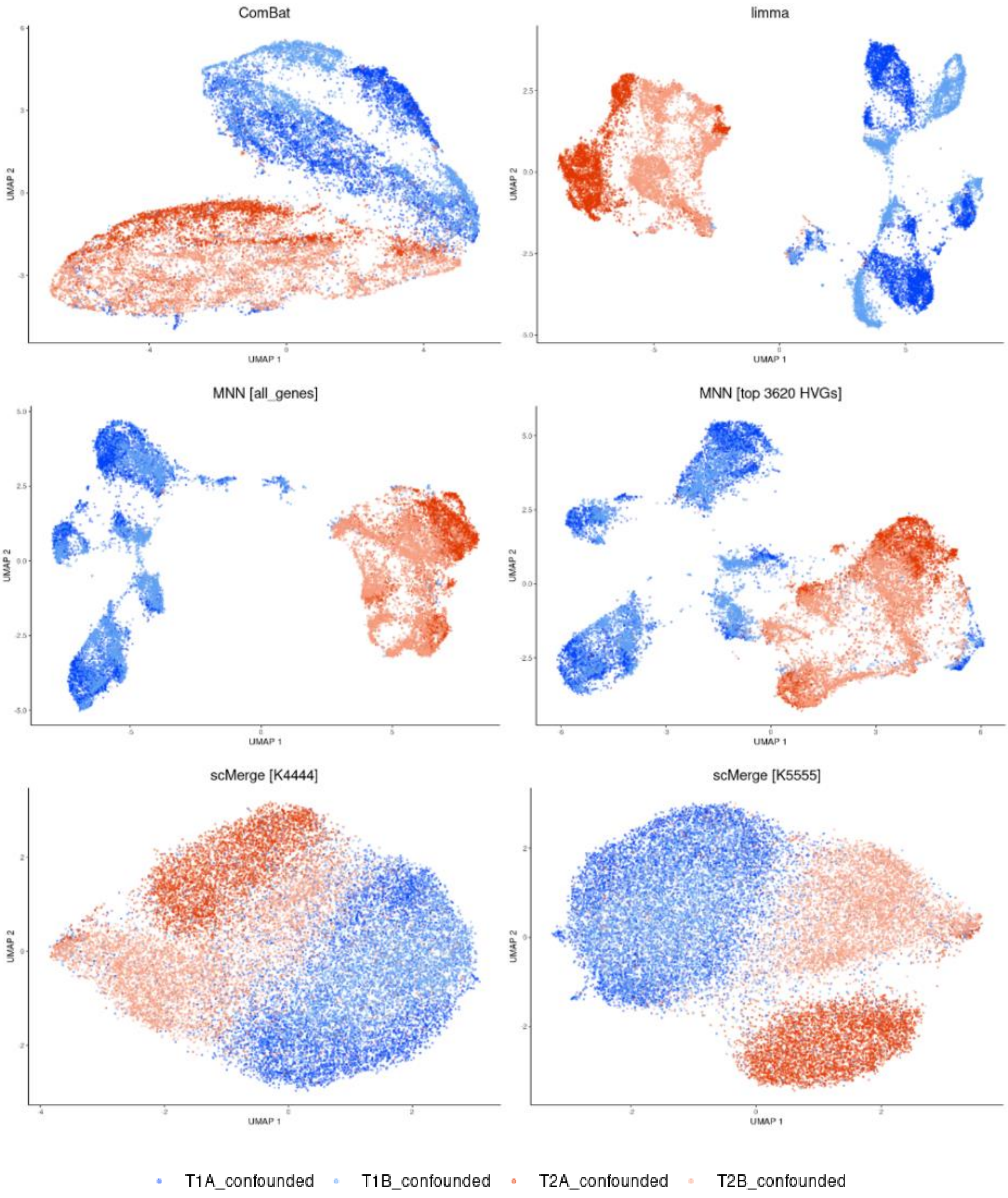


Figure 32. *UMAP plots after batch correction.*

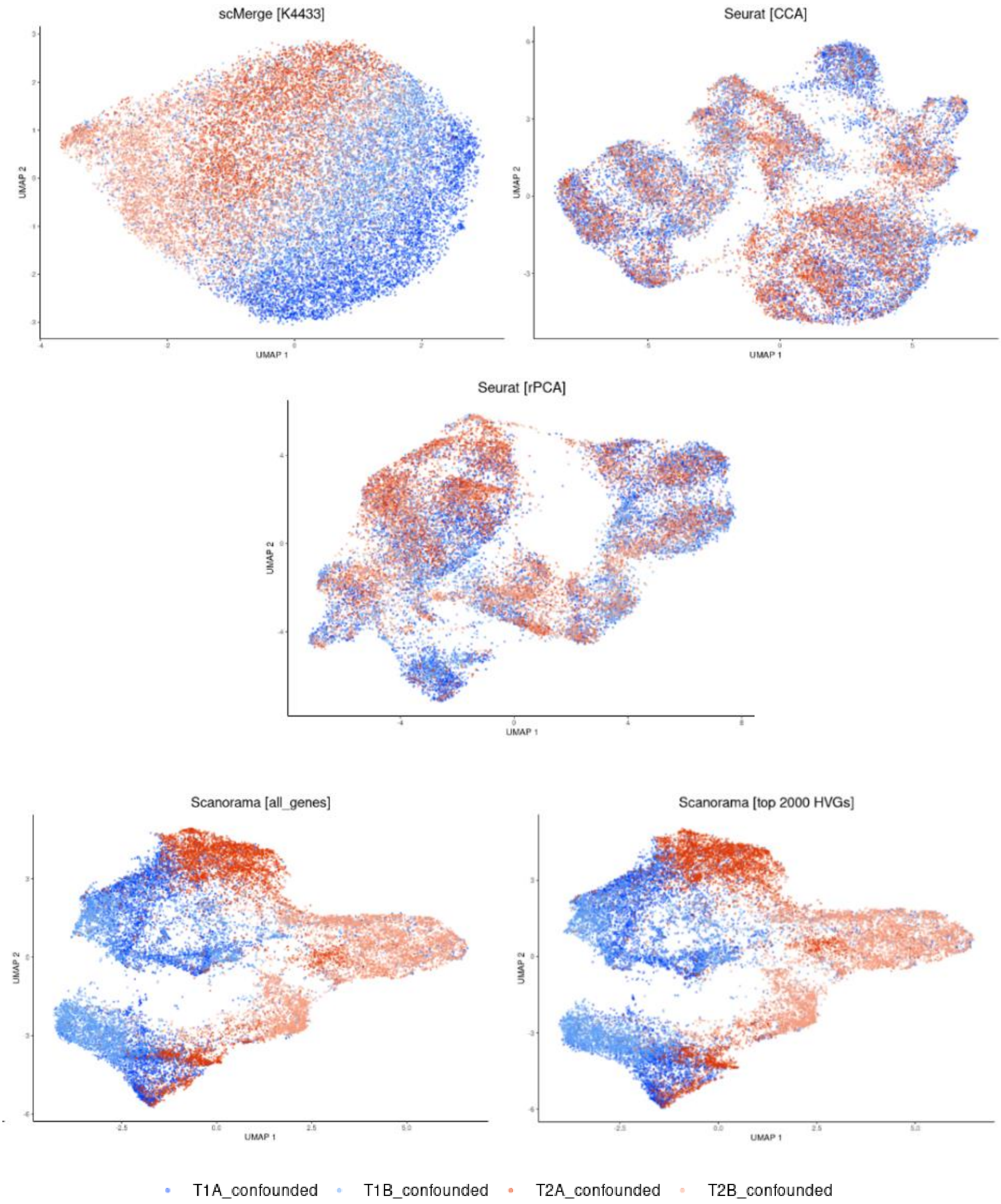


Figure 32. *continued*

To evaluate the impact of batch-effect correction on feature-level layer of the original data, the distribution of feature-level statistics (mean, variance, detection rate) was determined before (**Figure 33**) and after the correction (**Figure 34 - 35**).

The range of corrected mean expression values is lower compared to the original data, and genes with originally high mean expression are no longer ‘visible’ on the histograms from the corrected matrix **(Figure 34)**. This effect is particularly evident in Scanorama. However, ComBat-seq is an exception to this rule where the original range was preserved. It was the only method that preserved both the count nature and the original distribution of the data. Moreover, for MNN and Scanorama negative values started to occur in the corrected matrix, which makes biological interpretation difficult.

The correction also distorts the mean-variance relationship that is characteristic of the scRNAseq data. There is a sharp collapse of the log variance in the upper range of the mean expression, indicating that genes with higher average expression no longer follow the distribution of the raw data (**Figure 35** – left column).

The relationship between average expression and detection rate is conserved only for ComBat-seq and MNN (**Figure 35** – right column). Limma introduces small expression values to all cells for many low expression genes (dropout rate equal 1), while scMerge and Scanorama consistently increase the dropout rate with increased expression of the gene.

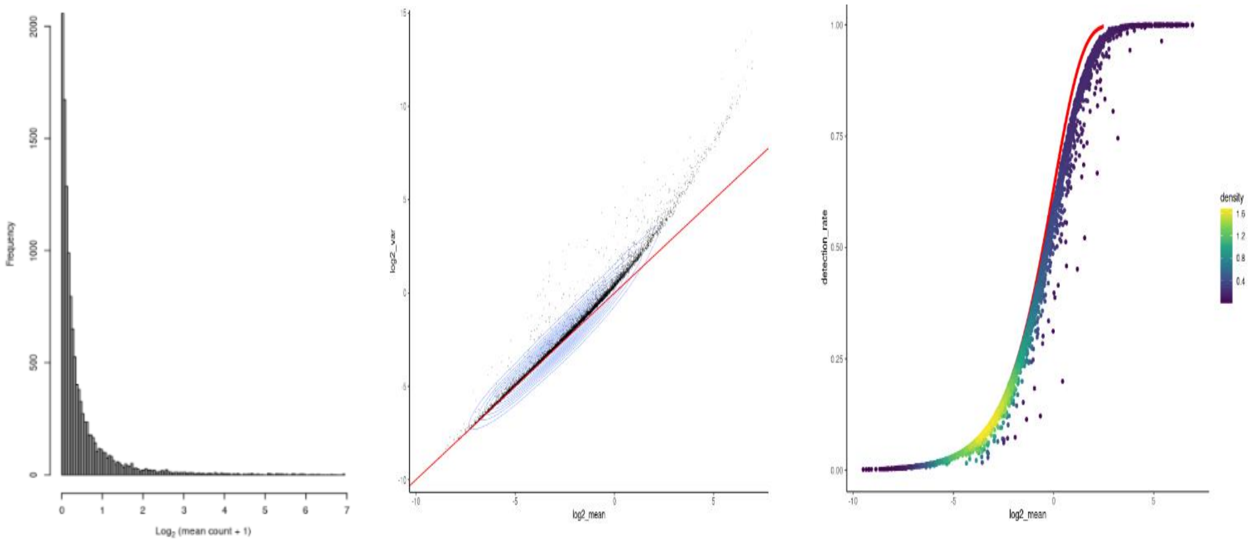


Figure 33*. Feature characteristics of confounded study before batch effect correction (for sample T1A). From the left: (i) histogram of average gene expression, (ii) scatter plot of variance vs mean expression (red line with intercept = 0 and slope = 1) and (iii) detection rate vs average expression (red line indicates the expected distribution under a Poisson model. Individual points are colored by the number of neighboring points).*

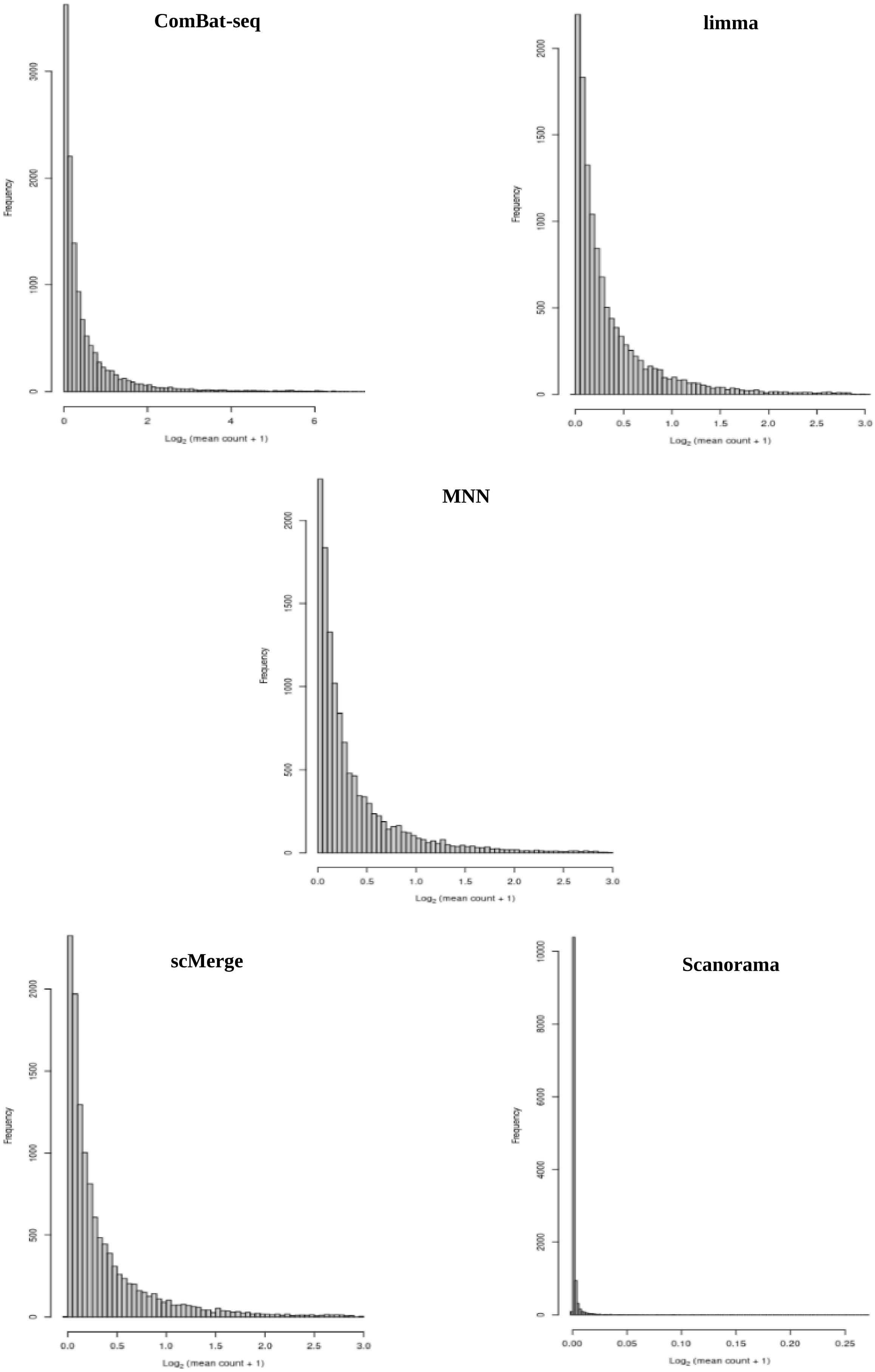
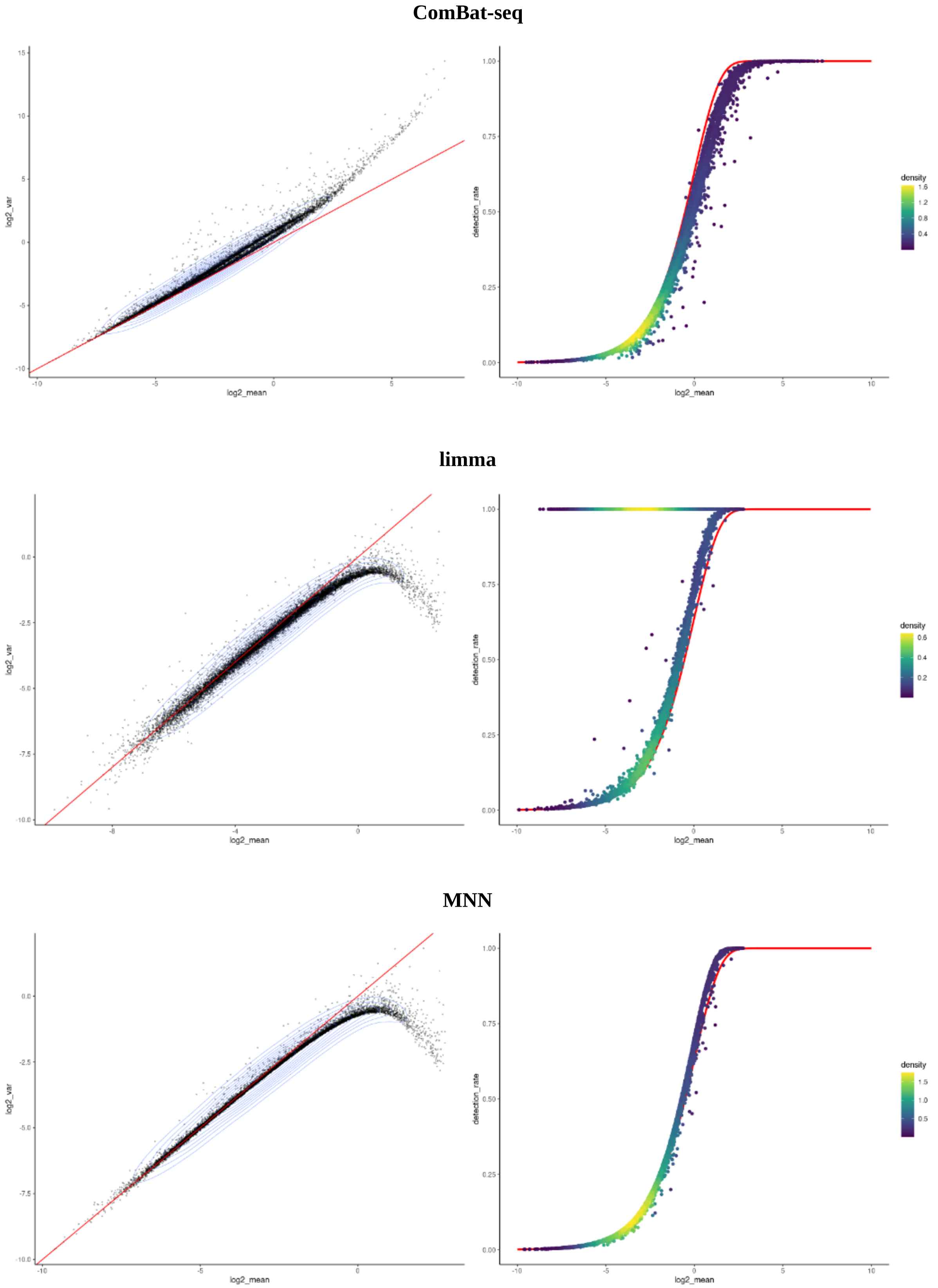


***Figure 34**. Histograms of average gene expression after batch effect correction (for sample T1A).*

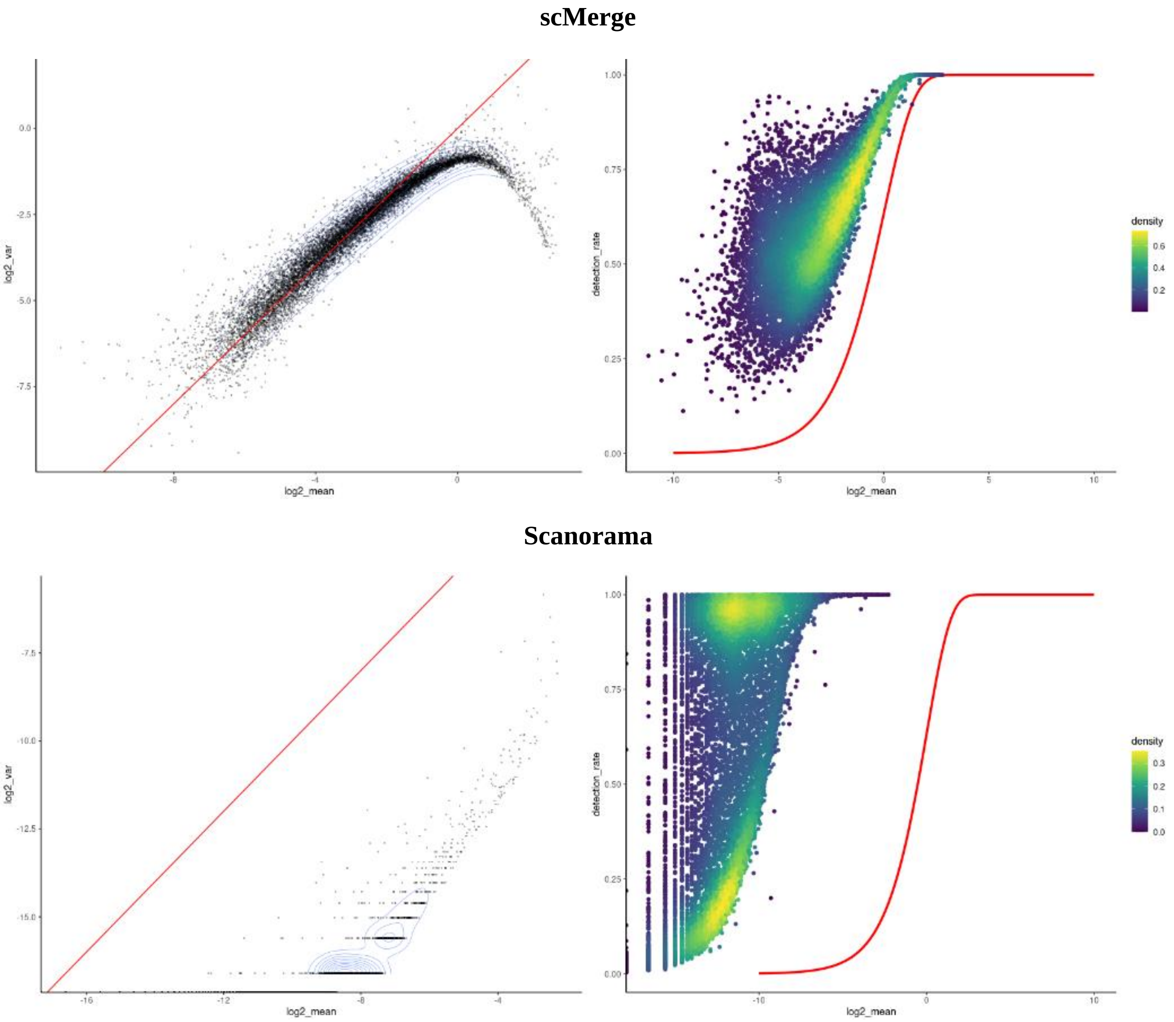


ComBat-seq

limma

MNN

Figure 35. *Feature level characteristics of confounded study after batch effect correction (for sample T1A). LEFT column: scatter plot of variance vs mean expression (red line with intercept = 0 and slope = 1); RIGHT column: detection rate vs average expression (red line indicates the expected distribution under a Poisson model. Individual points are colored by the number of neighboring points).*



scMerge

Scanorama

***Figure 35**. continued*

IV.2 Global noise filtering

All datasets exhibit a significant level of global noise, which is manifested in high dropout rates exceeding 90%. Many genes are rarely detected in any cell, with a dropout rate close to 1 (**Figure 36**).

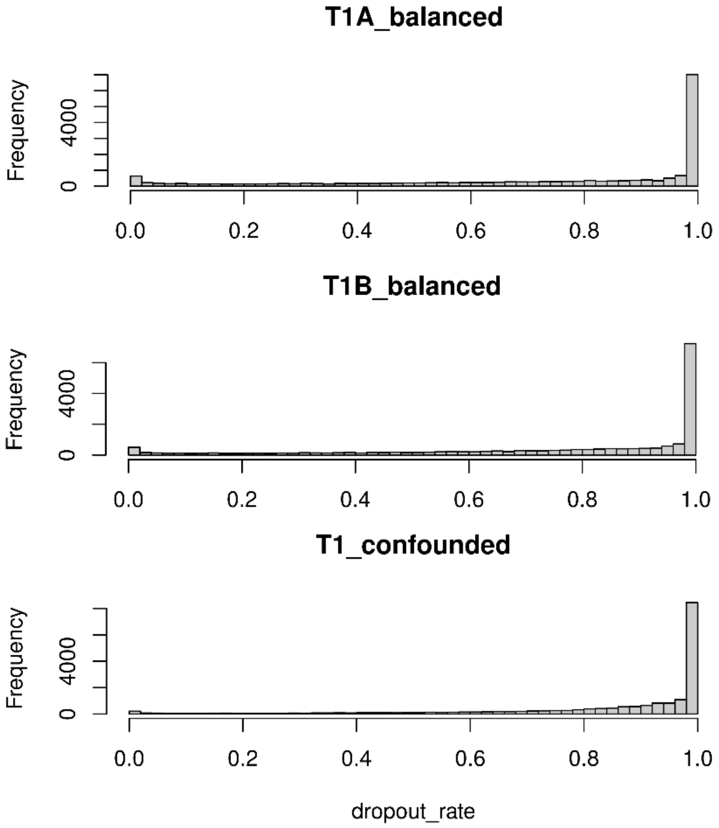


Figure 36. *Histograms of dropout rates before filtration. For clarity, only T1 timepoint from each dataset is shown, but the same level of dropouts was observed for all timepoints and datasets.*

To filter out these noisy genes in a data-driven manner, a method based on hierarchical clustering (HC) was proposed. Typically, the output of HC is in the form of a dendrogram. However, a dendrogram of several thousand genes would not provide informative results. Therefore, the results are presented in the form of a reader-friendly 'clustering tree' [136]. The clustering tree shows the relationships between clusters at multiple resolutions (K) (**Figure 37**). It allows the reader to examine how samples change their groupings as the number of clusters increases. As expected, genes with high dropout rates form a large cluster labeled as 1. Furthermore, their assignment to this cluster remains stable across different clustering resolutions. However, smaller subgroups of genes with lower dropout rates stand out from cluster 1 at different resolutions, which is desirable as we aim to discard only clearly redundant genes.

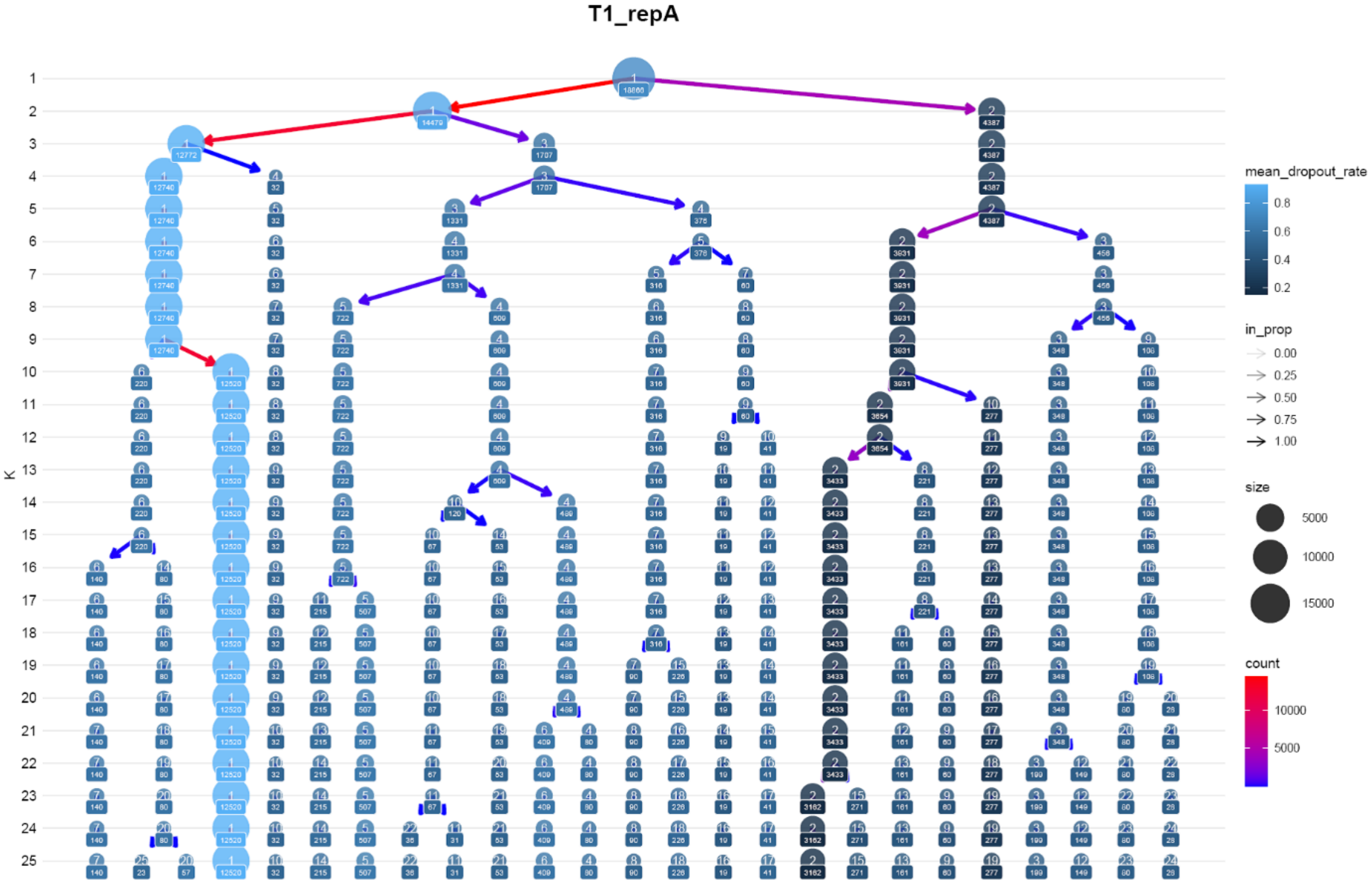


Figure 37. *Visualization of hierarchical clustering at different resolution (for T1 sample from repetition A). K – number of clusters (resolution). Nodes are colored according to the average dropout rate of the members and sized according to the number of features they represent (exact number is shown in label). Edges (arrows) are also colored according to the number of features. The transparency is adjusted according to the in-proportion, with stronger lines showing edges are more important to the higher-resolution cluster.*

Since the cluster assignments remained stable below K = 25, this value was established as a threshold for evaluating the distribution of dropout rates across all 25 clusters. It was observed that genes belonging to cluster 1 have noticeably higher dropout rates compared to genes in other clusters (**Figure 38**). Therefore, it is reasonable to discard genes in cluster 1 from further analysis. This process results in obtaining a reduced feature space for each sample/batch, which will be referred to as the 'reduced domain.' The number of genes before discarding noisy genes will be referred to as the 'full domain.'

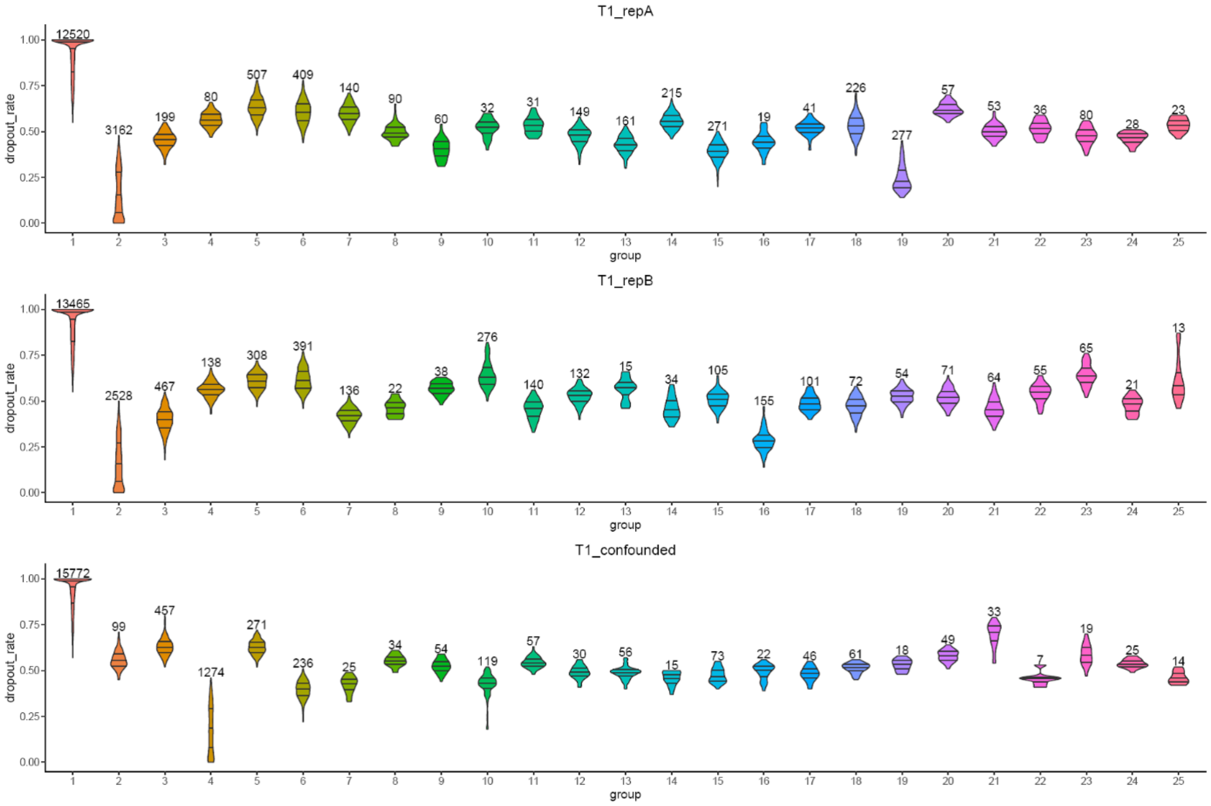


Figure 38. Distribution of dropout rates across clusters identified by hierarchical clustering at K = 25. For clarity only T1 sample from each dataset is shown.

Although there are still genes with zero expression after filtration, their proportion is substantially lower compared, and the distribution of dropout rates is more uniform compared to the full domain (**Figure 39**).

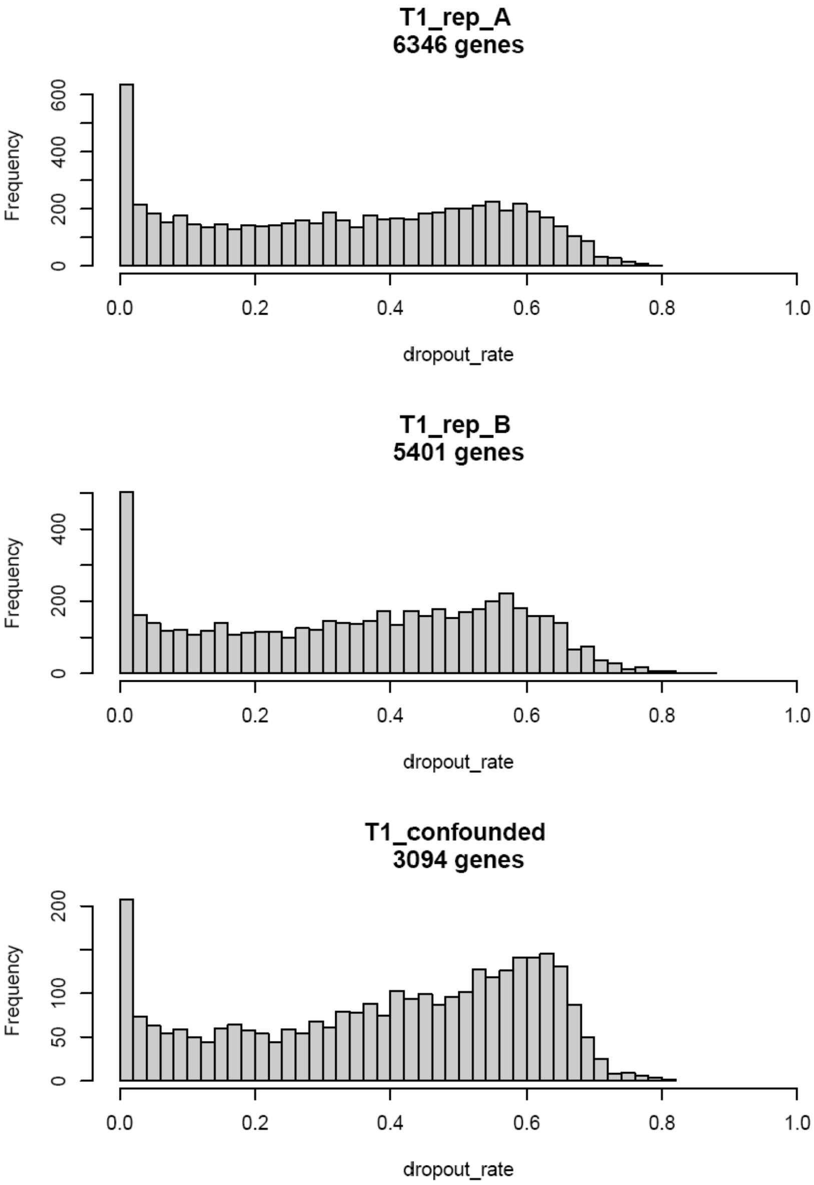


Figure 39. *Histograms of dropout rates after global noise filtration. There is a number of genes remaining after the global filtration indicated on the plot. T1 sample for each dataset is shown.*

A summary of the number of features before and after filtration is presented in **Table 2**.

Table 2. *The number of features left after global noise filtration.*

		reduced domain		
timepoint	full domain	**repA**	**repB**	**confounded**
T1	**18 866**	6 346	5 401	3 094
T2		6 253	6 713	3 226
T3		6 613	6 723	2 661

IV.3 Local noise filtering

The feature space obtained after global filtration, although significantly reduced, still contained genes that do not contribute much to the clustering process except for noise. Further optimization was performed locally (for each cluster discovered) based on variance decomposition into gaussian mixture components. Examples of the results from variance decomposition are presented on **Figure 40**. There were highly overlapping components observed, particularly for confounded dataset. Additionally, components covering almost the entire range of variance were observed for all datasets. However, they are not informative.

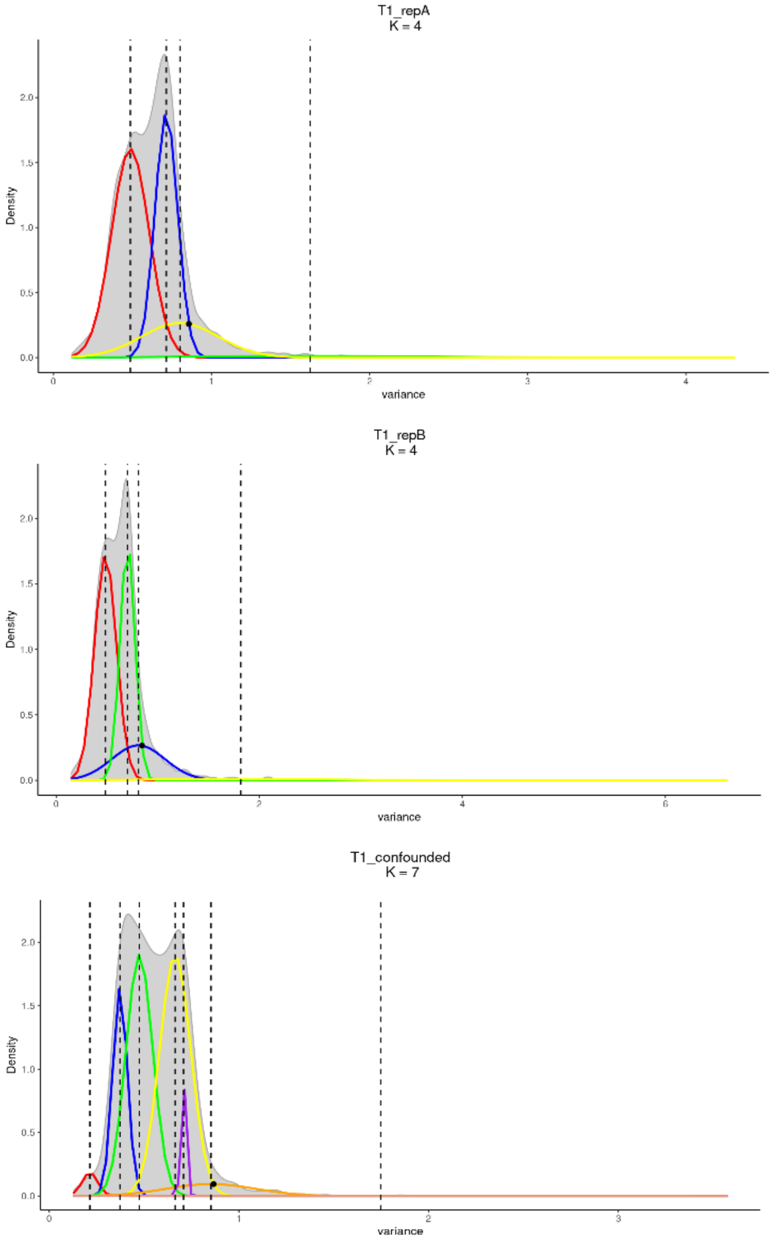


Figure 40. *Variance decomposition with GMM. Black dots indicate the threshold of variance above which genes are labeled as highly variable (HVG). The optimal number of components (K) were defined by BIC.*

IV.4 K-means clustering

To evaluate the impact of feature space filtration on the quality of clustering, k-means clustering (classic and sparse) was performed with three scenarios, with full, reduced and GMM filtered feature space (see Methods). The optimal number of clusters was determined by Calinski-Harabasz index, and it was equal to 2 for all samples and depth of analysis.

When the full domain is considered, the clusters are blurred due to the presence of many noise features that do not contribute to the clustering process (**Figure 41**). After global filtration (reduced domain), the quality of clustering substantially improved. The clusters became more distinct, except in T3 where no improvement was observed. With GMM filtered domain (HVGs), the quality of clustering did not improve compared to the reduced domain; instead, it remained at the same level. However, it must be noted that HVGs represent only a small portion of the genes in the reduced domain.

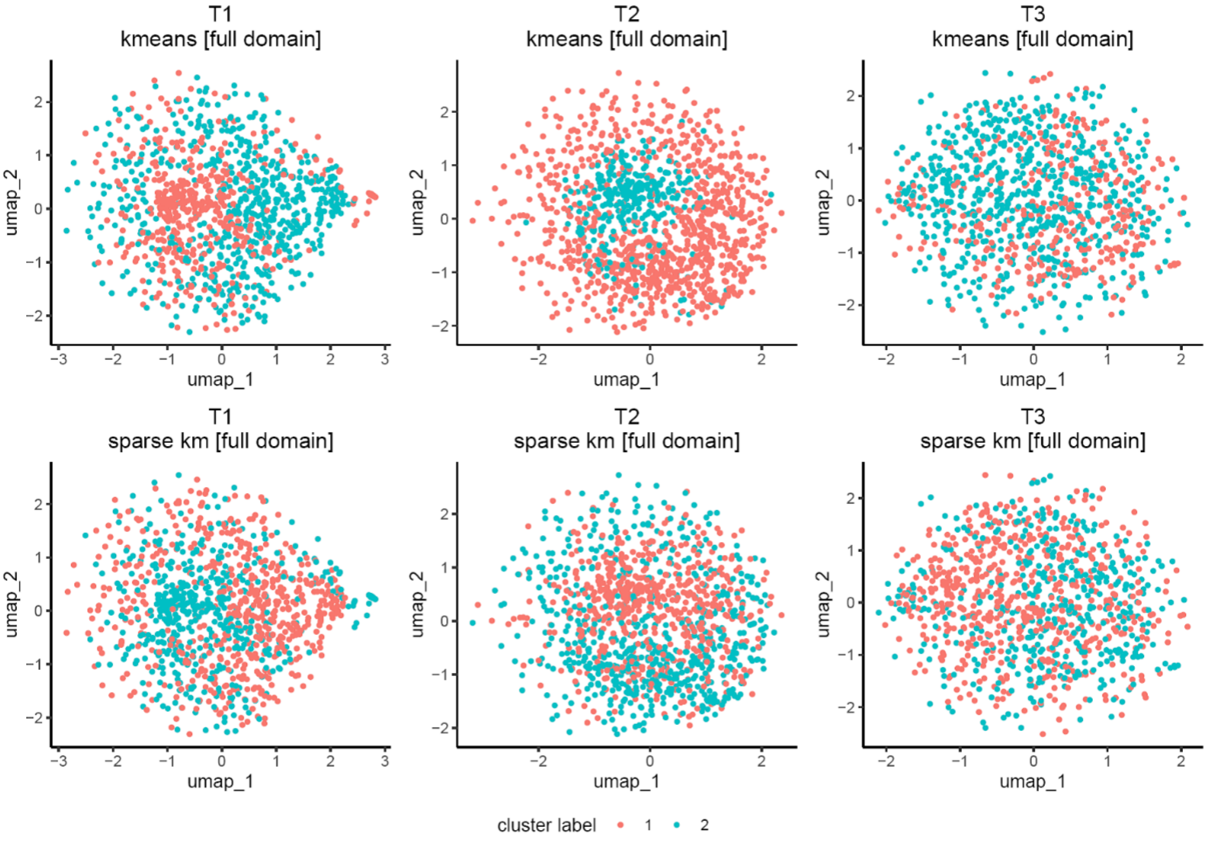


Figure 41. *K-means clustering with different scenarios. UMAP plots for repetition A of balanced study. Two variants of k-means clustering were involved: classic k-means, referred to as kmeans, and sparse k-means, referred to as sparse km.*

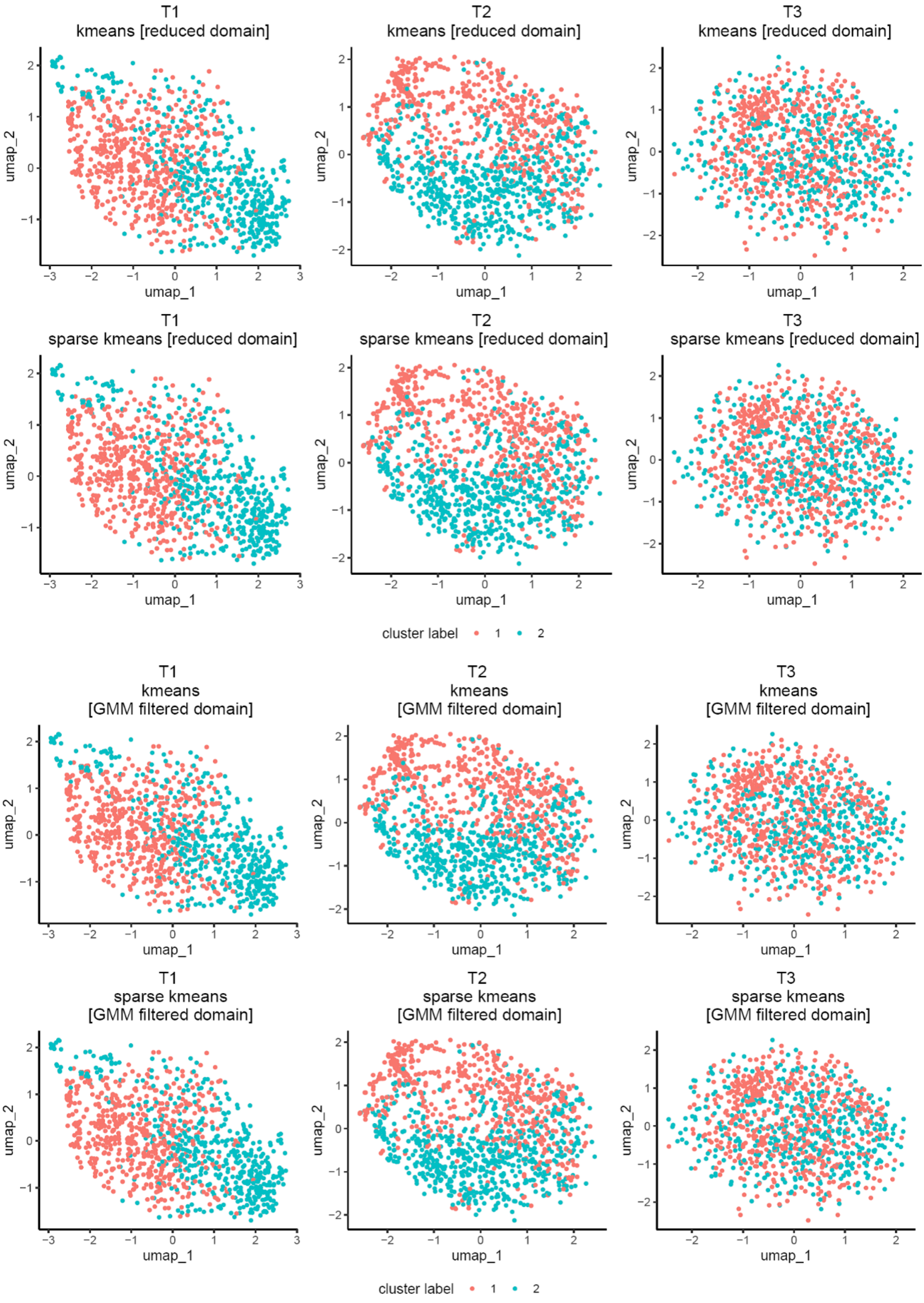


Figure 41. *continued*

The next step was to evaluate whether the weights assigned by sparse k-means were higher for the highly variable genes (HVGs) compared to the non-HVG group. This evaluation aimed to verify the proposed GMM-based feature selection method. The distribution of weights was assessed in both groups for repetitions A and B of the balanced study. In both cases, the weights in the HVG group were substantially higher, despite comprising only a small number of genes, constituting 2-3% of the full domain (**Figure 42**).

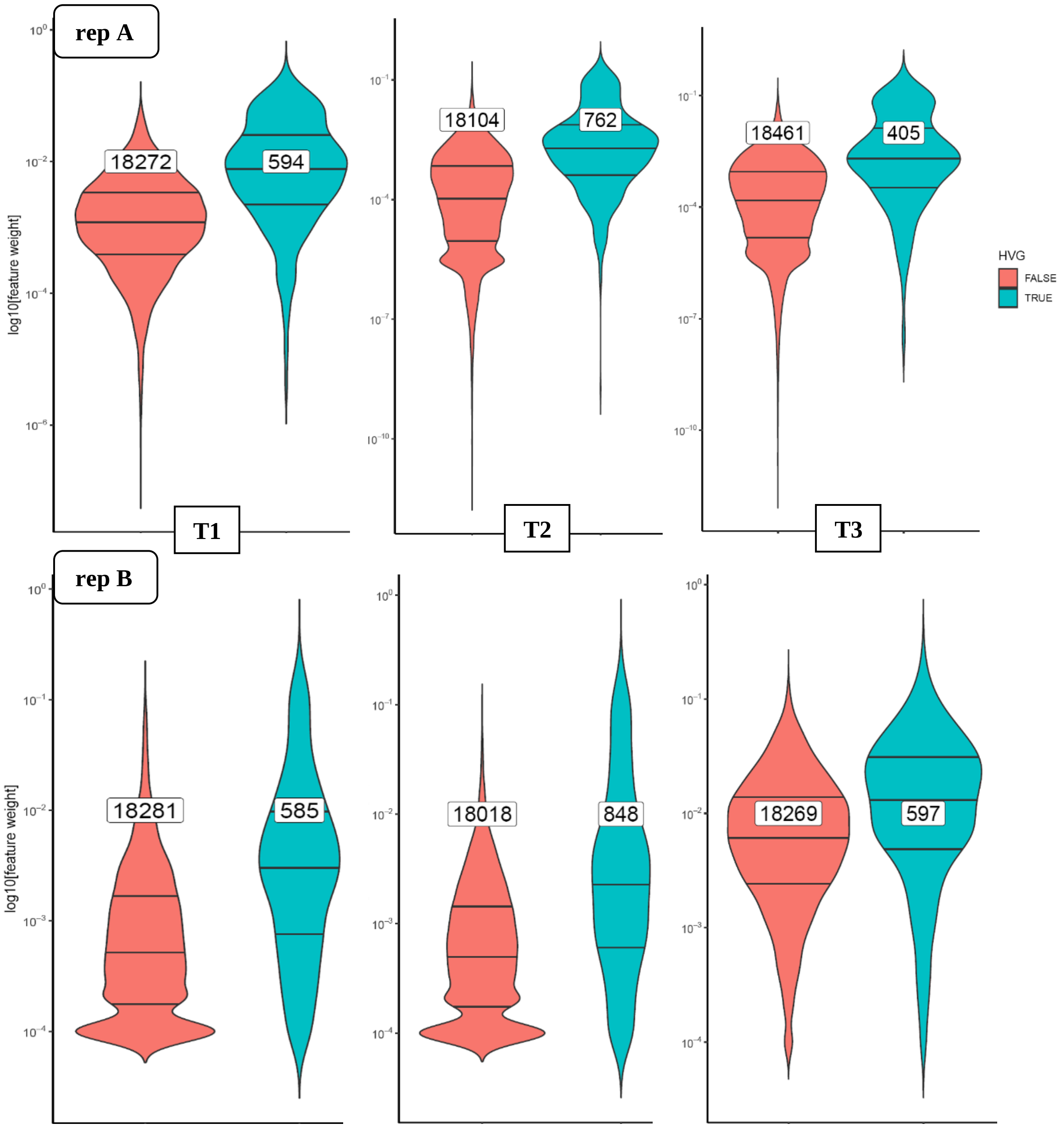


Figure 42. *Distribution of feature weights assigned by sparse k-means. The evaluation was performed for both repetitions of balanced study. Two groups were considered: involving only HVGs obtained by GMM filtering (green), and non-HVG group (red). Weights were assigned automatically by the algorithm. There is a number of features depicted inside of each violin plot.*

To delve deeper into the mechanism of weight assignment, quadrant analysis was conducted. This analysis examined the relationship between gene variances and assigned weights in each quadrant using Pearson and Spearman correlation. Similar to previous step, both repetitions of the balanced dataset were analyzed. The threshold limit for the X and Y-axes was determined based on the median value in the respective axis parameter. This division resulted in the formation of four quadrants.

Particular attention was given to quadrants arranged diagonally: Q1/Q3 and Q2/Q4. The former corresponds to a scenario where high weights are assigned to genes with high variance (Q1) and conversely low weights are assigned to genes with low variance (Q3). This is the most desired scenario in terms of evaluation of the feature selection strategy. Q2/Q4 reflects the opposite situation, where low weights are assigned to genes with high variances (Q2) or high weights are assigned to genes with low variances (Q4).

In repetition A, a majority of highly variable genes were present in Q1 and only a portion in Q2 (**Figure 43**) At least three times higher values of both correlation coefficients were observed for each timepoint in Q1 compared to Q2. High correlations were observed also for Q4 quadrant (high weight – low variance). However, the number of genes in Q4 was almost 5 times smaller compared to Q1 which result of obtaining a spuriously large correlation coefficient.

In repetition B, the median value of assigned weights was equal to 0, hence only two quadrants were obtained. In both of them the correlations are similar (**Figure 44**).

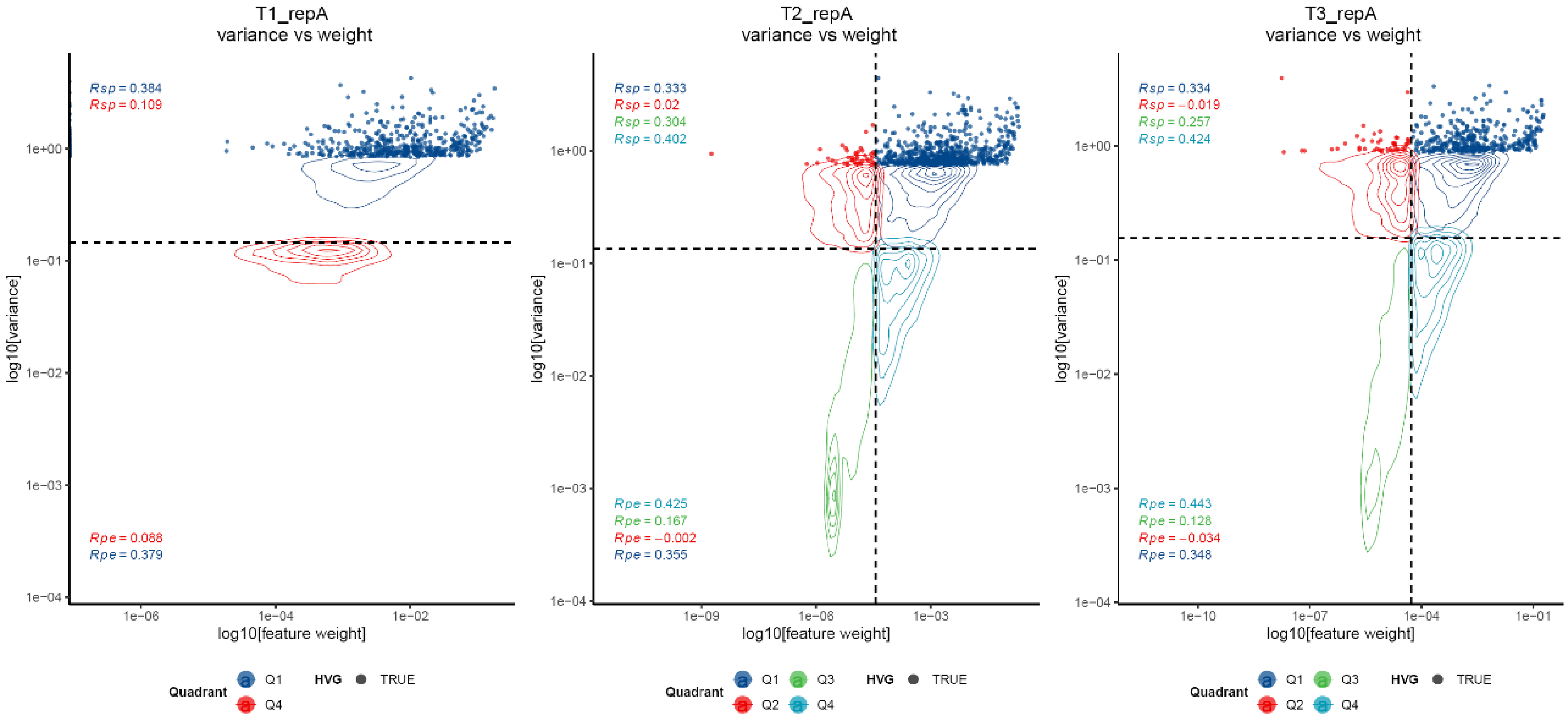


Figure 43. *Analysis of weights assigned by sparse k-means algorithm (repetition A). Rsp – Spearman's correlation coefficient, Rpe – Pearson's correlation coefficient, HVG – highly variable gene identified by GMM variance decomposition. The dashed line represents the median value of the weight on the x-axis and variance on the y-axis. When not plotted, the median value of the respective parameter is equal zero.*

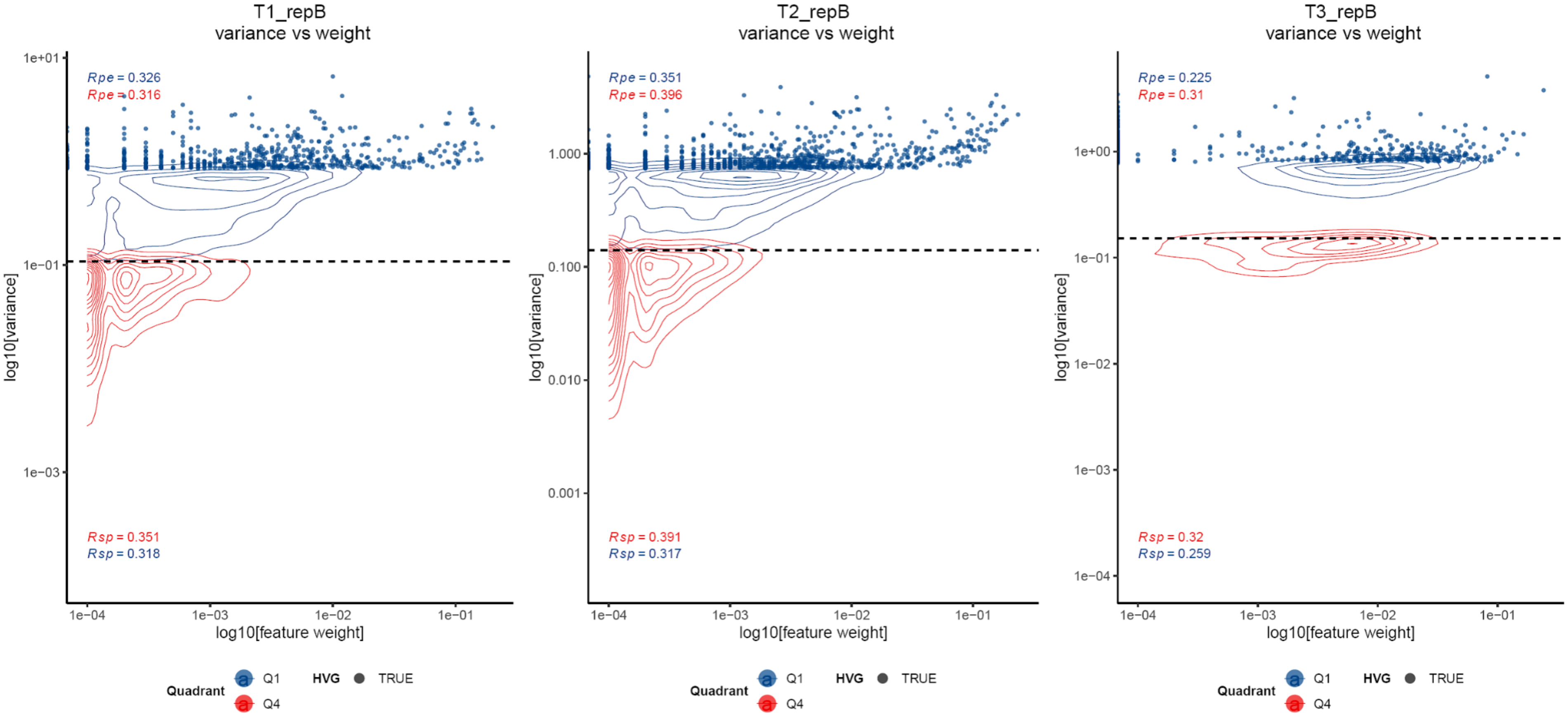


Figure 44. *Analysis of weights assigned by sparse k-means algorithm (repletion B). Rsp – Spearman's correlation coefficient, Rpe – Pearson's correlation coefficient, HVG – highly variable gene identified by GMM variance decomposition. The dashed line represents the median value of the weight on the x-axis and variance on the y-axis. When not plotted, the median value of the respective parameter is equal zero.*

IV.5 Functional analysis and cluster linkage

IV.5.1 Evaluation of cluster-specific pathways

This section involves comparisons between the repetition A of the balanced study, considered as a reference dataset, and the confounded dataset. After performing the GSVA analysis, the enrichment score was obtained for 111 pathways in repetition A and 106 for repetition B while the confounded dataset only included 50 pathways.

The first step was to identify cluster-specific pathways for each dataset. To achieve this, the Cliff's delta effect size metric was utilized. Since the clusters were evaluated in a one-to-others scenario, the pathways with the highest value of this metric for a particular cluster would dominate over the other pathways and should be considered cluster-specific.

The Cliff's delta ranging from -0.2 to 0.3, were observed across all clusters discovered in repetition A of the balanced study. Most values clustered around 0, and the histograms tend to be left-skewed. However, only small differences were observed between the distributions of Cliff's delta statistics (**Figure 45**). Conversely, in confounded dataset stronger differences between histograms were observed (**Figure 46**).

The top three cluster-specific pathways with the highest (head) and lowest (tail) effect sizes are listed in **Table 3** for the reference dataset and **Table 4** for the confounded dataset. For convenience, pathways with the highest value of the effect size metric will be referred to as 'top3_high', while those with the lowest value of Cliff’s delta statistics will be referred to as 'top3_low'.

The most frequently occurring pathways in both reference and confounded dataset were the ‘progesterone-mediated oocyte maturation’ and ‘cell cycle’. These pathways often appeared together and were observed within ‘top3_high’ group. However, in cluster T1_II_1 (reference) these pathways were present in the ‘top3_low’ group as well as for T1_I_1, T2_II_1 and T3_II_1 cluster in the confounded study.

The ‘progesterone-mediated oocyte maturation’ pathway is involved in the maturation of oocytes in females and is regulated by the hormone progesterone. Additionally, several identified pathways were related to cell cycle regulation, which is crucial for cell growth, division, and replication. These pathways included:

- The pathway involved in the metabolism of purine nucleotides which are necessary for DNA replication and cell growth during the cell cycle. This pathway was identified in clusters T2_II_1 and T3_I_1.
- The ‘Base excision repair pathway’, which is involved in DNA repair mechanism that corrects damaged or modified bases in the DNA molecule to maintain genomic integrity.
- The ‘Mismatch repair’, which corrects errors that occur during DNA replication, such as mismatches and small insertions or deletions.
- The ‘Notch signalling’ pathway, which interacts with key cell cycle regulators such as cyclins, cyclin-dependent kinases (CDKs), and inhibitors of CDKs (CDKIs) to control cell cycle transitions and ensure proper cell proliferation.
- The ‘p53 signaling pathway’: its main member is p53 protein which regulates cell cycle arrest, cellular senescence, or apoptosis.
- The ‘TGF-Beta signaling’, which can enhance proliferation and promote tumor progression.

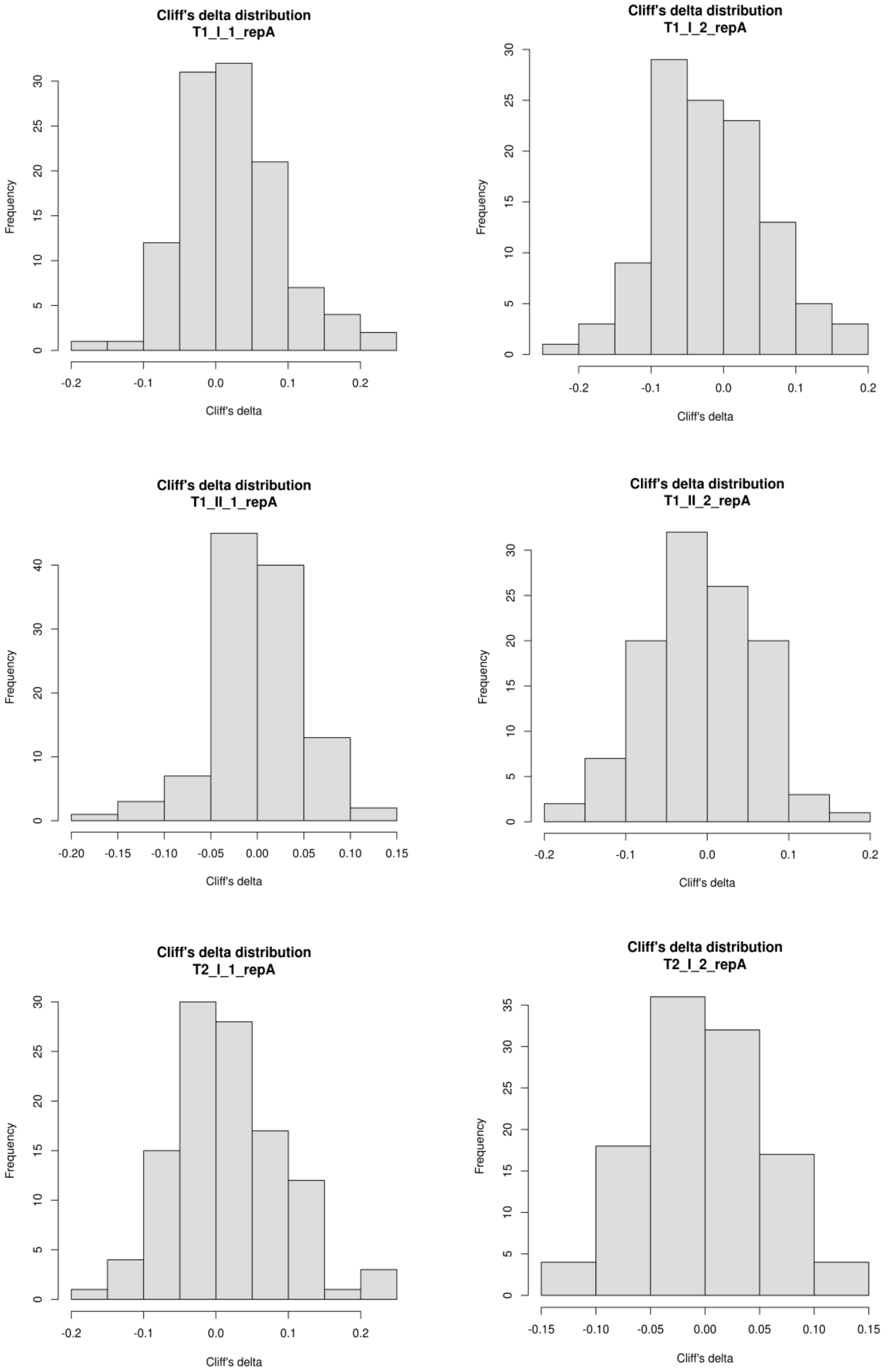
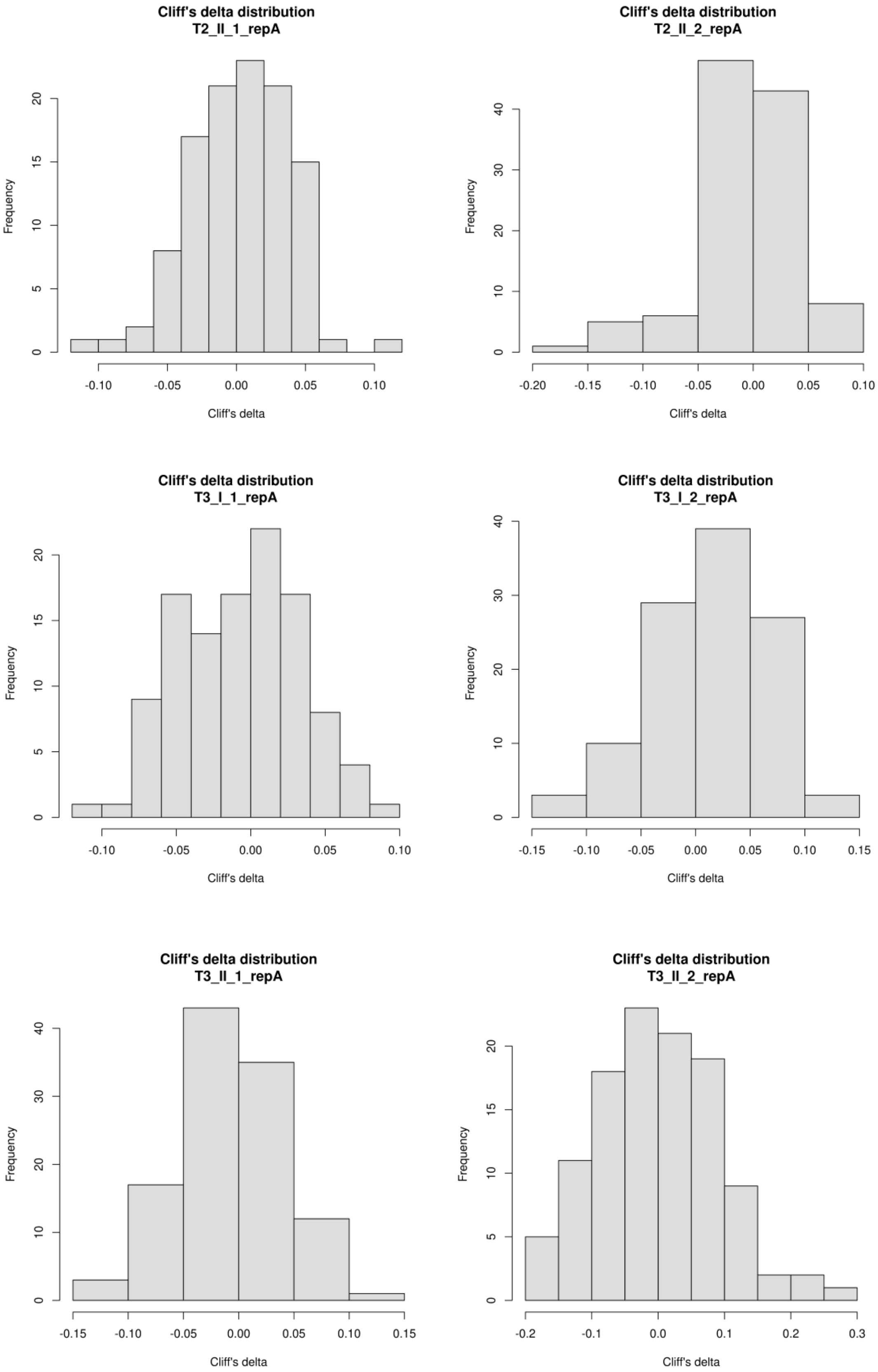


Figure 45. *Distribution of Cliff's delta effect size statistics across pathways in repetition A of balanced study.*



Cliff's delta distribution T2_II_1_repA

Cliff's delta distribution T2_II_2_repA

Cliff's delta distribution T3_I_1_repA

Cliff's delta distribution T3_I_2_repA

Cliff's delta distribution T3_II_1_repA

Cliff's delta distribution T3_II_2_repA

Figure 45. *continued*

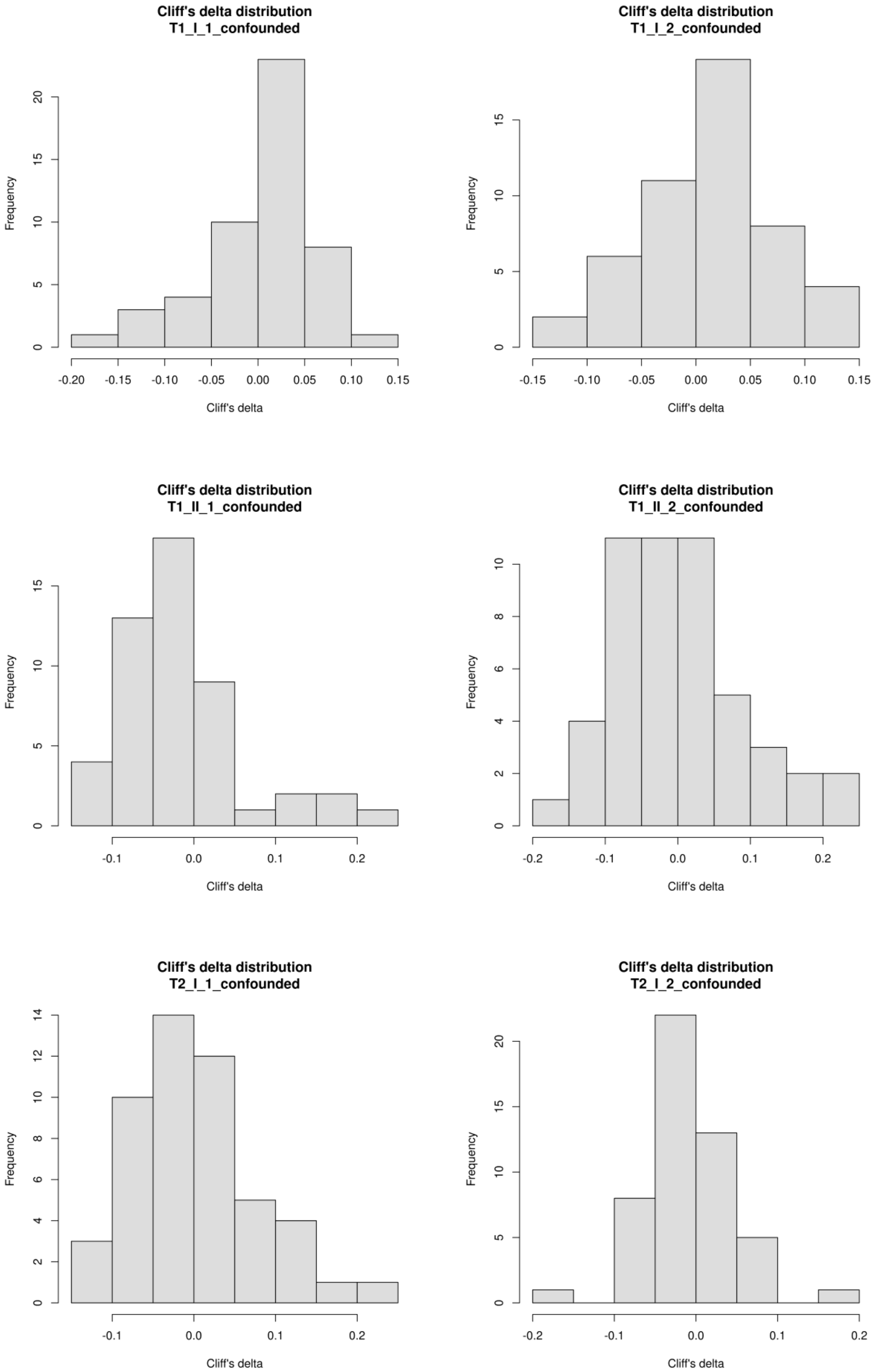


Figure 46. *Distribution of Cliff's delta effect size statistics across pathways in confounded study.*

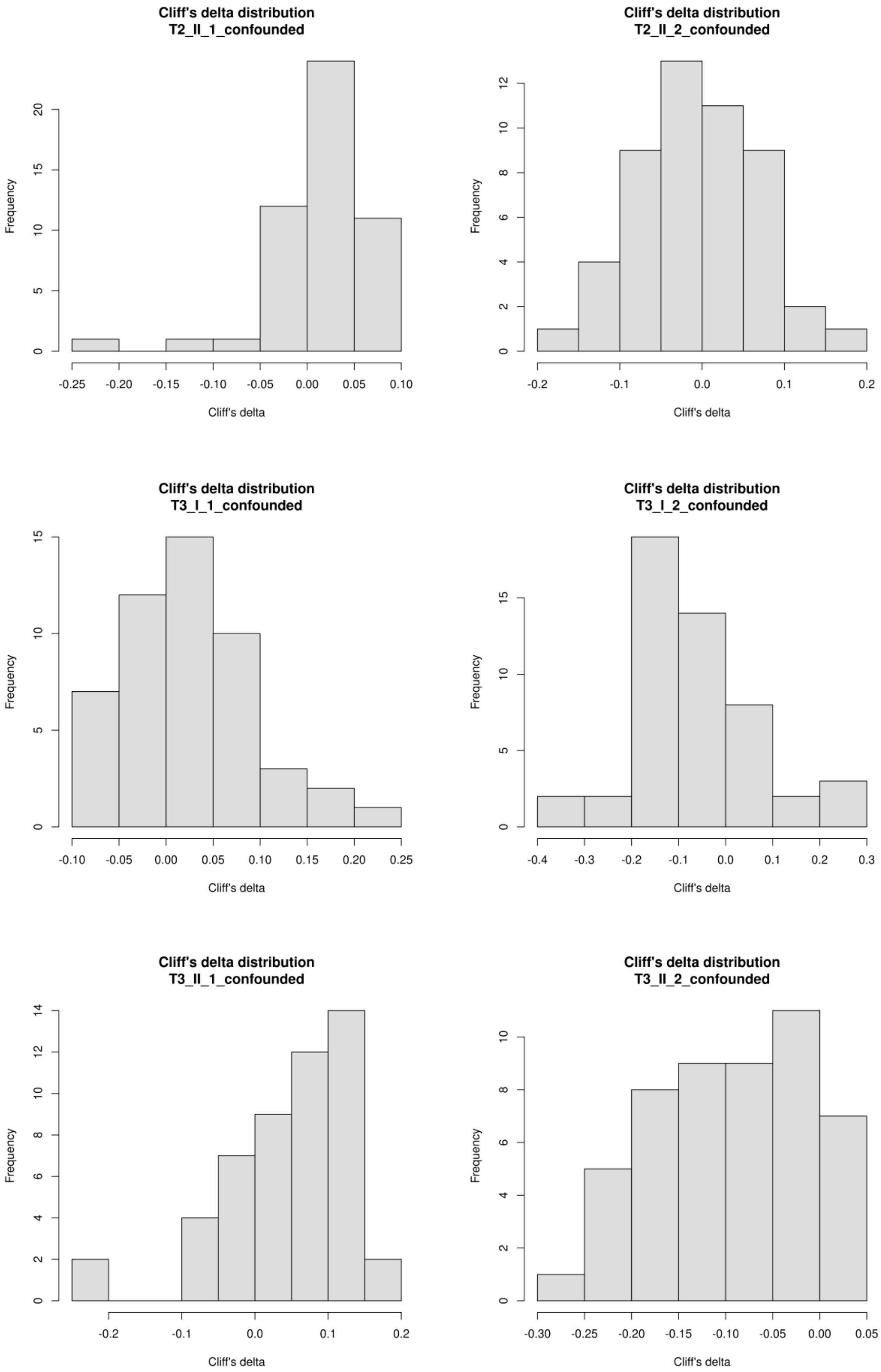


Figure 46. *continued*

Table 3.Top three pathways with the highest/lowest effect size (ES) for each cluster in balanced study

cluster	pathway	ES
T1_I_1	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.236
	CELL_CYCLE	0.211
	OOCYTE_MEIOSIS	0.183
	PARKINSONS_DISEASE	-0.081
	SPLICEOSOME	-0.127
	SNARE_INTERACTIONS_IN_VESICULAR_TRANSPORT	-0.153
T1_I_2	CELL_CYCLE	0.188
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.162
	DNA_REPLICATION	0.156
	FATTY_ACID_METABOLISM	-0.153
	CYTOKINE_CYTOKINE_RECEPTOR_INTERACTION	-0.186
	ECM_RECEPTOR_INTERACTION	-0.202
T1_II_1	GLYCEROLIPID_METABOLISM	0.104
	SNARE_INTERACTIONS_IN_VESICULAR_TRANSPORT	0.103
	BASE_EXCISION_REPAIR	0.092
	OOCYTE_MEIOSIS	-0.117
	CELL_CYCLE	-0.145
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	-0.198
T1_II_2	ECM_RECEPTOR_INTERACTION	0.161
	NOTCH_SIGNALING_PATHWAY	0.118
	CELL_ADHESION_MOLECULES_CAMS	0.114
	DNA_REPLICATION	-0.136
	BASE_EXCISION_REPAIR	-0.158
	CELL_CYCLE	-0.171
T2_I_1	CELL_CYCLE	0.250
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.222
	SMALL_CELL_LUNG_CANCER	0.205
	CYTOKINE_CYTOKINE_RECEPTOR_INTERACTION	-0.140
	FC_GAMMA_R_MEDIATED_PHAGOCYTOSIS	-0.146
	DRUG_METABOLISM_OTHER_ENZYMES	-0.162
T2_I_2	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.144
	CELL_CYCLE	0.140
	SMALL_CELL_LUNG_CANCER	0.115
	OXIDATIVE_PHOSPHORYLATION	-0.108
	HUNTINGTONS_DISEASE	-0.110
	PARKINSONS_DISEASE	-0.122

pathways with the highest ES are marked in red

Table 3. continued

cluster	pathway	ES
T2_II_1	PURINE_METABOLISM	0.103
	PEROXISOME	0.062
	RENAL_CELL_CARCINOMA	0.056
	MELANOGENESIS	-0.067
	WNT_SIGNALING_PATHWAY	-0.093
	JAK_STAT_SIGNALING_PATHWAY	-0.114
T2_II_2	CYTOKINE_CYTOKINE_RECEPTOR_INTERACTION	0.075
	SNARE_INTERACTIONS_IN_VESICULAR_TRANSPORT	0.072
	LYSOSOME	0.064
	CELL_CYCLE	-0.132
	PURINE_METABOLISM	-0.140
	SMALL_CELL_LUNG_CANCER	-0.156
T3_I_1	N_GLYCAN_BIOSYNTHESIS	0.092
	PURINE_METABOLISM	0.077
	GLYCEROPHOSPHOLIPID_METABOLISM	0.065
	BLADDER_CANCER	-0.074
	CHRONIC_MYELOID_LEUKEMIA	-0.084
	P53_SIGNALING_PATHWAY	-0.102
T3_I_2	LEUKOCYTE_TRANSENDOTHELIAL_MIGRATION	0.136
	HYPERTROPHIC_CARDIOMYOPATHY_HCM	0.122
	DILATED_CARDIOMYOPATHY	0.111
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	-0.120
	SPHINGOLIPID_METABOLISM	-0.125
	CELL_CYCLE	-0.136
T3_II_1	CYTOSOLIC_DNA_SENSING_PATHWAY	0.106
	AMINO_SUGAR_AND_NUCLEOTIDE_SUGAR_METABOLISM	0.093
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.090
	GLUTATHIONE_METABOLISM	-0.101
	PURINE_METABOLISM	-0.105
	GLYCEROLIPID_METABOLISM	-0.137
T3_II_2	CELL_CYCLE	0.289
	MISMATCH_REPAIR	0.223
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.207
	LYSOSOME	-0.183
	HYPERTROPHIC_CARDIOMYOPATHY_HCM	-0.191
	SNARE_INTERACTIONS_IN_VESICULAR_TRANSPORT	-0.198

Table 4. Top three pathways with the highest/lowest effect size (ES) for each cluster in confounded study

cluster	pathway	ES
T1_I_1	SMALL_CELL_LUNG_CANCER	0.122
	MAPK_SIGNALING_PATHWAY	0.092
	CHRONIC_MYELOID_LEUKEMIA	0.074
	OOCYTE_MEIOSIS	-0.114
	CELL_CYCLE	-0.117
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	-0.159
T1_I_2	HUNTINGTONS_DISEASE	0.113
	OXIDATIVE_PHOSPHORYLATION	0.109
	ALZHEIMERS_DISEASE	0.103
	UBIQUITIN_MEDIATED_PROTEOLYSIS	-0.093
	CHRONIC_MYELOID_LEUKEMIA	-0.109
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	-0.115
T1_II_1	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.249
	OOCYTE_MEIOSIS	0.158
	CELL_CYCLE	0.154
	PARKINSONS_DISEASE	-0.114
	OXIDATIVE_PHOSPHORYLATION	-0.121
	HUNTINGTONS_DISEASE	-0.122
T1_II_2	TGF_BETA_SIGNALING_PATHWAY	0.239
	CELL_CYCLE	0.231
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.182
	VIBRIO_CHOLERAE_INFECTION	-0.119
	SPLICEOSOME	-0.125
	PROTEASOME	-0.155
T2_I_1	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.214
	CELL_CYCLE	0.190
	TGF_BETA_SIGNALING_PATHWAY	0.149
	ANTIGEN_PROCESSING_AND_PRESENTATION	-0.106
	ENDOCYTOSIS	-0.111
	PROTEIN_EXPORT	-0.117
T2_I_2	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.188
	OOCYTE_MEIOSIS	0.097
	CELL_CYCLE	0.080
	HUNTINGTONS_DISEASE	-0.068
	TGF_BETA_SIGNALING_PATHWAY	-0.076
	DNA_REPLICATION	-0.168

pathways with the highest ES are marked in red

Table 4. continued

cluster	pathway	ES
T2_II_1	ADHERENS_JUNCTION	0.080
	REGULATION_OF_ACTIN_CYTOSKELETON	0.079
	FOCAL_ADHESION	0.072
	OOCYTE_MEIOSIS	-0.093
	CELL_CYCLE	-0.122
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	-0.205
T2_II_2	DNA_REPLICATION	0.153
	NUCLEOTIDE_EXCISION_REPAIR	0.138
	LYSOSOME	0.105
	PATHWAYS_IN_CANCER	-0.112
	OOCYTE_MEIOSIS	-0.118
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	-0.151
T3_I_1	CELL_CYCLE	0.205
	P53_SIGNALING_PATHWAY	0.178
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.170
	HUNTINGTONS_DISEASE	-0.074
	OXIDATIVE_PHOSPHORYLATION	-0.075
	PARKINSONS_DISEASE	-0.078
T3_I_2	CELL_CYCLE	0.284
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	0.256
	PROTEASOME	0.210
	VIBRIO_CHOLERAE_INFECTION	-0.206
	CALCIUM_SIGNALING_PATHWAY	-0.326
	APOPTOSIS	-0.366
T3_II_1	REGULATION_OF_ACTIN_CYTOSKELETON	0.161
	CALCIUM_SIGNALING_PATHWAY	0.157
	APOPTOSIS	0.148
	GAP_JUNCTION	-0.095
	CELL_CYCLE	-0.213
	PROGESTERONE_MEDIATED_OOCYTE_MATURATION	-0.213
T3_II_2	GLUTATHIONE_METABOLISM	0.040
	PROTEASOME	0.030
	TGF_BETA_SIGNALING_PATHWAY	0.028
	CELL_CYCLE	-0.243
	CALCIUM_SIGNALING_PATHWAY	-0.244
	OOCYTE_MEIOSIS	-0.253

IV.5.2 Within dataset cluster linkage

Pairwise comparisons of Cliff's delta effect size statistics across clusters were performed to link clusters between batches (timepoints) within the dataset. In the first approach, similarities between clusters were evaluated through Pearson correlation coefficient.

Repetition A of the reference study: 7 pairs exhibited large correlations ($|r| \geq 0.7$). Within the same timepoint, a negative correlation was observed for pairs, while a positive correlation was observed mainly between clusters from different timepoints (**Figure 47**). The strongest positive relationship was observed for the following pairs: T1_II_2 – T3_I_2 (r = 0.748) and for T1_I_1 – T3_II_2 (r = 0.732).

Repetition B of the reference study: 2 pairs of highly correlated clusters were observed involving one positive relationship between clusters T2_I_2 and T3_I_2 (r = 0.720). From the medium correlation range, the highest positive was observed for T2_II_1 - T2_II_2 (**Figure 48**).

Confounded study: 11 highly correlated pairs were observed, including 4 positives, but these pairs differed from the reference. Nonetheless, negative correlations occurred mainly between clusters within the same timepoint (**Figure 49**).

In the second approach, similarities between clusters were evaluated using a metric called the "similarity score," which is simply the dot product of two vectors. Both the similarity score and Pearson correlation are related; however, the former focuses on representing the alignment between vectors, while the latter represents the strength and direction of the linear relationship between the variables. A larger dot product indicates a stronger alignment.

To visually track clusters across timepoints, the clusters were compared in the following manner: clusters from T1 were compared with clusters from T2, and clusters from T2 were compared with clusters from T3. Sankey diagrams were then generated for each dataset based on the two similarity metrics under consideration. The Sankey plot effectively illustrates the flow of clusters from T1 to T3 (**Figure 50**).

Pearson correlation:

In repetition A of the balanced study, one of the most active flow paths was observed from T1_I_1, passing through T2_I_2, and reaching T3_II_2 (**Figure 50** – top panel**)**. Similarly, in repetition B of the balanced study, the same target (T3_II_2) and intermediate cluster (T2_I_2) were observed, although the path started from T1_I_2 in this dataset. On the other hand, in the confounded dataset, the flow distribution was more uniform, with T2_I_1 and T3_I_1 receiving the highest flow amounts.

Similarity score:

When considering the similarity score measure, the overall flow structure is generally similar to that of the correlation-based metric. However, there is an improvement in resolution. For example, in repetition A, the most active path still begins at T1_I_1, but now the intermediate cluster has changed to T2_I_1, resulting in a more pronounced flow. (**Figure 50** – bottom panel) The target cluster, however, remains the same as in the correlation metric.

The Sankey diagrams reveal that a particular cluster can exhibit similarities with more than one other cluster. Based on maximization approach, only clusters with the highest positive similarity metric (separately for correlation and similarity score) were paired, allowing each cluster to form only one pair. (**Table 5** and **Table 6**). These pairs of clusters can be considered the most similar between timepoints (batches).

In the next section, comparisons will be made between corresponding timepoints of the reference and confounded datasets. To avoid unnecessary confusion, the results presented below will only focus on the similarity score.

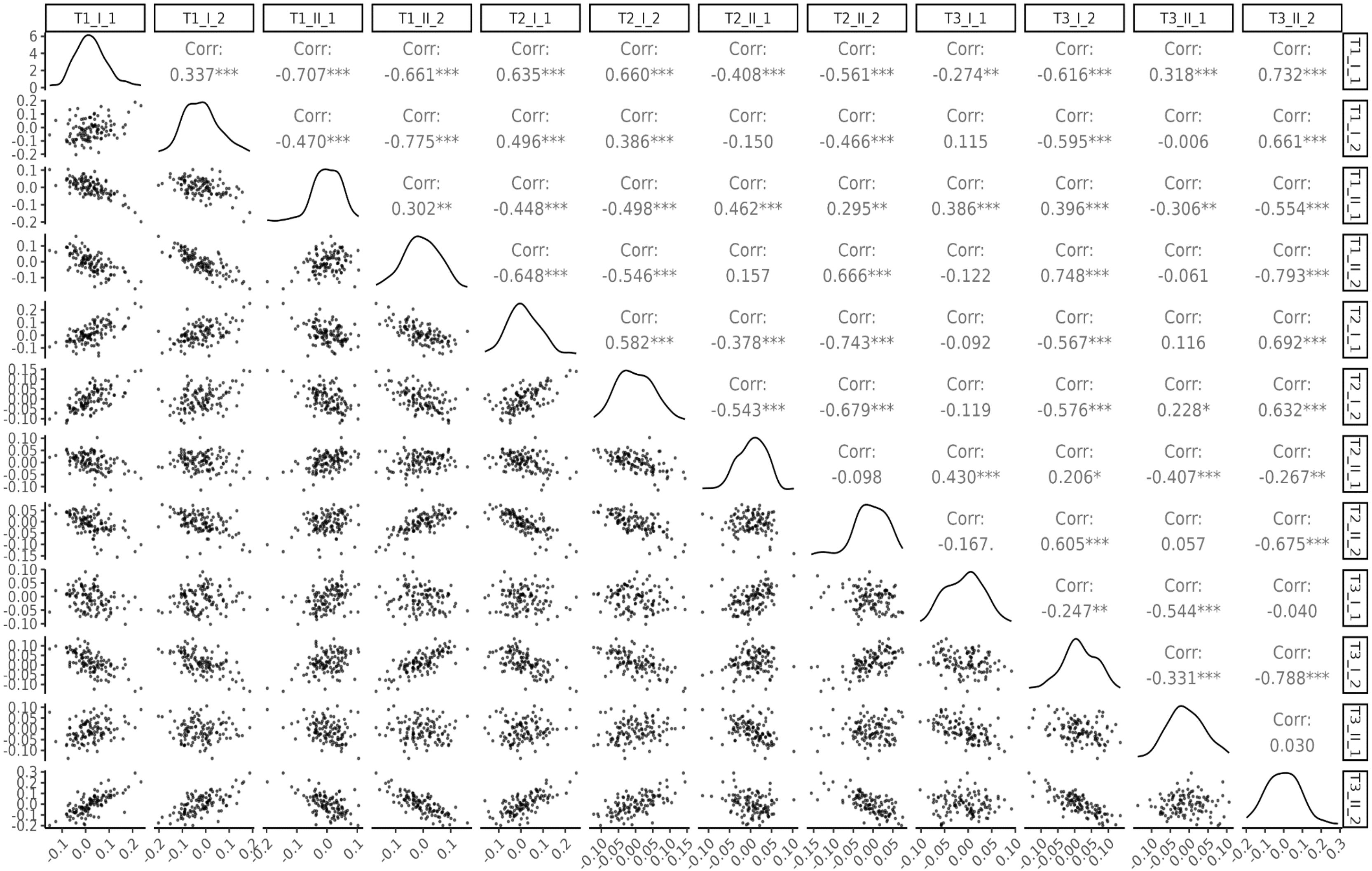


Figure 47. *Pairwise plots for balanced study (repetition A). Scatterplots of each pair are visualized on the left side of the plot (points represent value of Cliff's delta for each pathway), while the corresponding Pearson correlation value and significance are displayed on the right side. The diagonal represents the distribution of Cliff's delta values across pathways for each cluster. The significance of correlation is indicated by: "***" - if the p-value is < 0.001; "**" - if the p-value is < 0.01; "*" - if the p-value is < 0.05; "." - if the p-value is < 0.10 and "" – otherwise.*

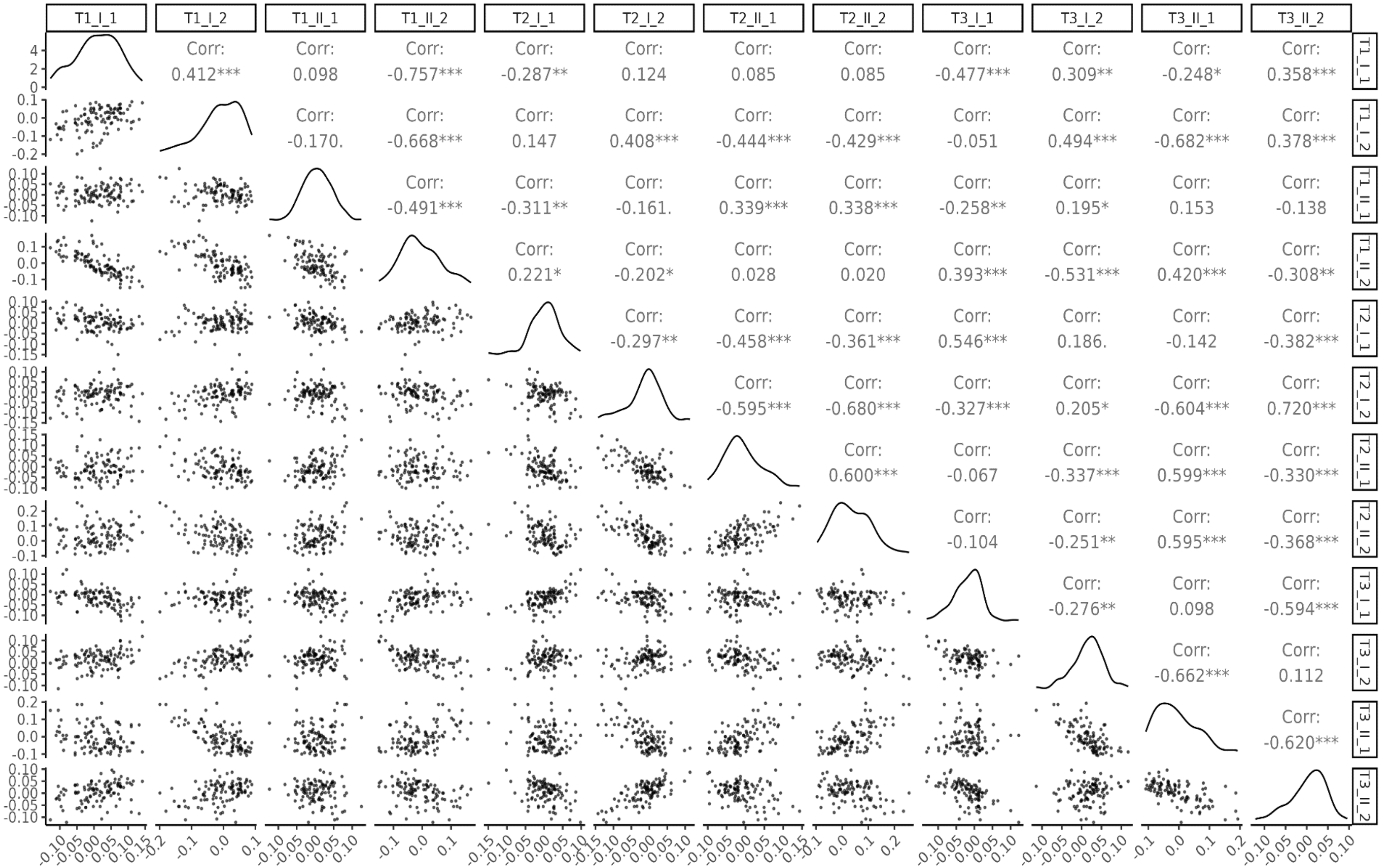


Figure 48. *Pairwise plots for balanced study (repetition B). Scatterplots of each pair are visualized on the left side of the plot (points represent value of Cliff's delta for each pathway), while the corresponding Pearson correlation value and significance are displayed on the right side. The diagonal represents the distribution of Cliff's delta values across pathways for each cluster. The significance of correlation is indicated by: "***" - if the p-value is < 0.001; "**" - if the p-value is < 0.01; "*" - if the p-value is < 0.05; "." - if the p-value is < 0.10 and "" – otherwise.*

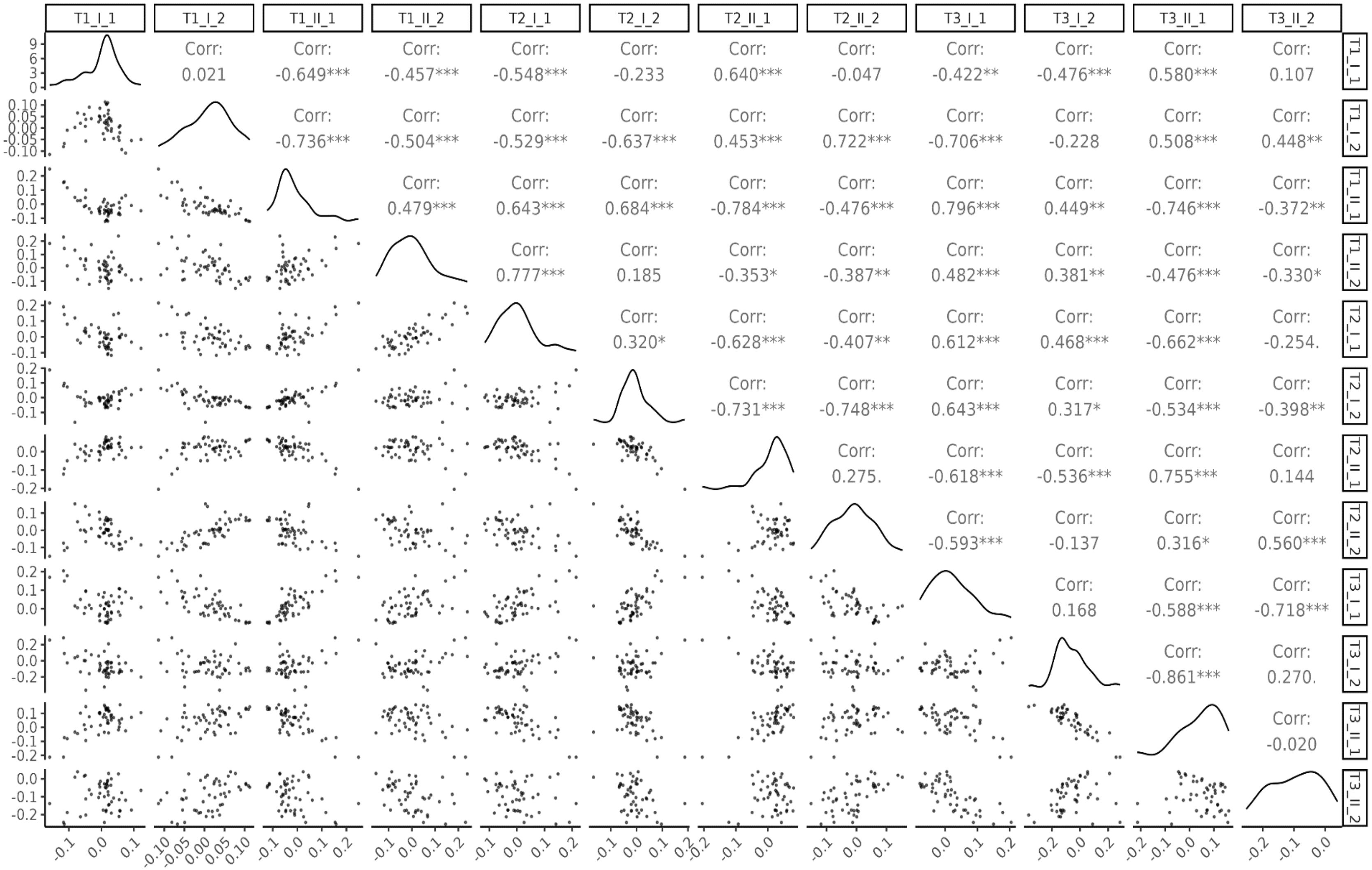


*Figure 49. Pairwise plots for confounded study. Scatterplots of each pair are visualized on the left side of the plot (points represent value of Cliff's delta for each pathway), while the corresponding Pearson correlation value and significance are displayed on the right side. The diagonal represents the distribution of Cliff's delta values across pathways for each cluster. The significance of correlation is indicated by: "***" - if the p-value is < 0.001; "**" - if the p-value is < 0.01; "*" - if the p-value is < 0.05; "." - if the p-value is < 0.10 and "" – otherwise.*

Table 5. *Pairs of clusters with the maximum values of correlation similarity.*

repA balanced			repB balanced			confounded		
cluster_T1	cluster_T2	max_corr	cluster_T1	cluster_T2	max_corr	cluster_T1	cluster_T2	max_corr
T1_II_2	T2_II_2	0.666	T1_I_2	T2_I_2	0.408	T1_II_2	T2_I_1	0.777
T1_I_1	T2_I_2	0.660	T1_II_1	T2_II_1	0.339	T1_I_2	T2_II_2	0.722
T1_I_2	T2_I_1	0.496	T1_II_2	T2_I_1	0.221	T1_II_1	T2_I_2	0.684
T1_II_1	T2_II_1	0.462	T1_I_1	T2_II_2	0.085	T1_I_1	T2_II_1	0.640

repA balanced			repB balanced			confounded		
cluster_T2	cluster_T3	max_corr	cluster_T2	cluster_T3	max_corr	cluster_T2	cluster_T3	max_corr
T2_I_1	T3_II_2	0.692	T2_I_2	T3_II_2	0.720	T2_II_1	T3_II_1	0.755
T2_II_2	T3_I_2	0.605	T2_II_1	T3_II_1	0.599	T2_I_2	T3_I_1	0.643
T2_II_1	T3_I_1	0.430	T2_I_1	T3_I_1	0.546	T2_II_2	T3_II_2	0.560
T2_I_2	T3_II_1	0.228				T2_I_1	T3_I_2	0.468

Table 6. *Pairs of clusters with the maximum values of similarity score.*

repA balanced			repB balanced			confounded		
cluster_T1	cluster_T2	max_sim_score	cluster_T1	cluster_T2	max_sim_score	cluster_T1	cluster_T2	max_sim_score
T1_I_1	T2_I_1	0.398	T1_I_2	T2_I_2	0.140	T1_II_2	T2_I_1	0.262
T1_II_2	T2_II_2	0.216	T1_II_1	T2_II_2	0.124	T1_II_1	T2_I_2	0.140
T1_I_2	T2_I_2	0.179	T1_II_2	T2_I_1	0.062	T1_I_2	T2_II_2	0.133
T1_II_1	T2_II_1	0.087	T1_I_1	T2_II_1	0.018	T1_I_1	T2_II_1	0.092

repA balanced			repB balanced			confounded		
cluster_T2	cluster_T3	max_sim_score	cluster_T2	cluster_T3	max_sim_score	cluster_T2	cluster_T3	max_sim_score
T2_I_1	T3_II_2	0.533	T2_II_2	T3_II_1	0.302	T2_I_1	T3_I_2	0.206
T2_II_2	T3_I_2	0.149	T2_I_2	T3_II_2	0.152	T2_II_1	T3_II_1	0.195
T2_I_2	T3_II_1	0.066	T2_I_1	T3_I_1	0.088	T2_II_2	T3_II_2	0.193
T2_II_1	T3_I_1	0.063				T2_I_2	T3_I_1	0.109

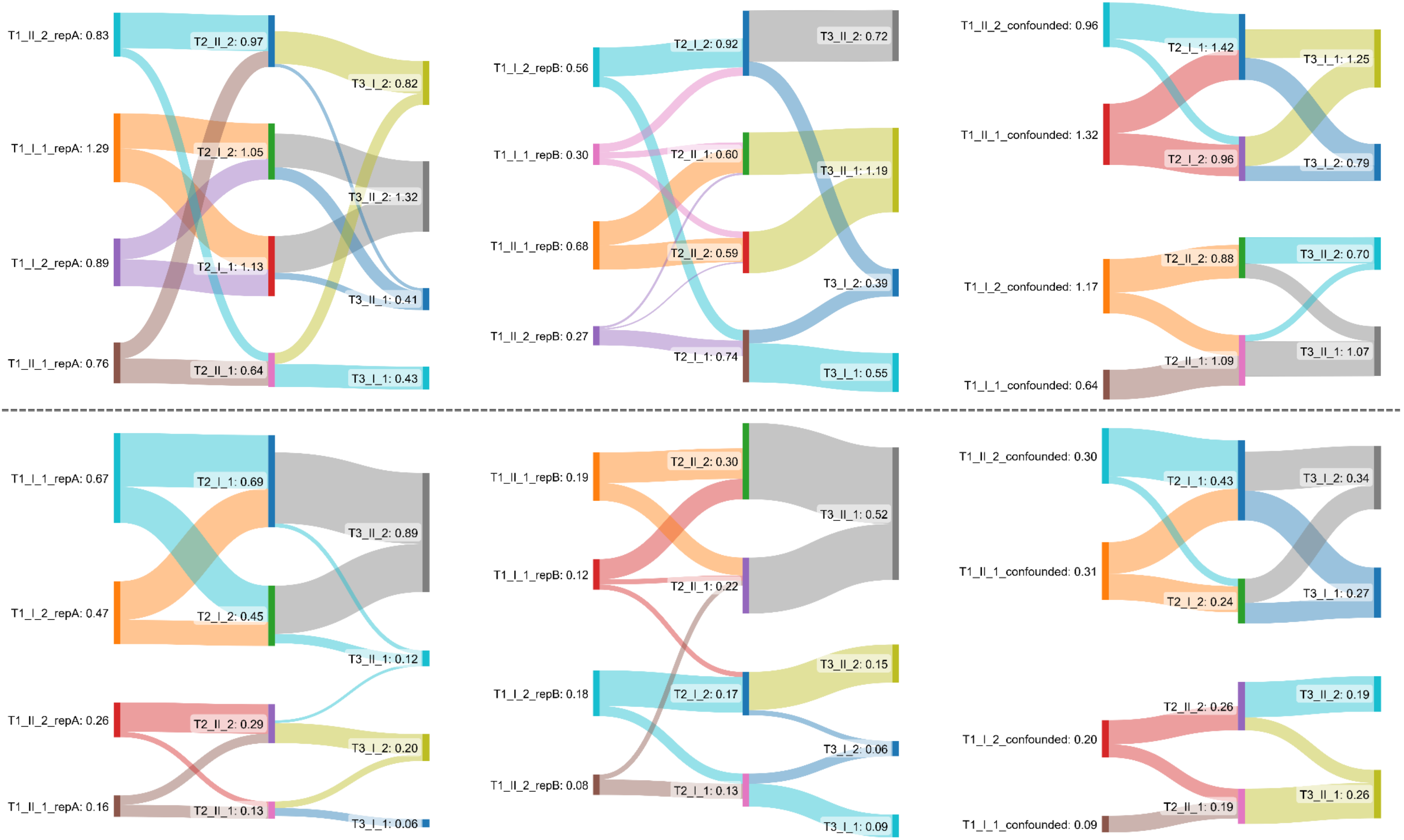


Figure 50. *Sankey diagrams for each dataset. Left – repA_balanced, middle – repB_balanced, and right – confounded dataset. Sankey plot presents how clusters flow from T1 to T3. The thickness of flows is proportional to the value of the metric considered (Pearson correlation – top row or similarity score – bottom row). The label on the right of the cluster name represents the sum of the values coming out of the given cluster (for source nodes) or the sum of the values coming in the given cluster (for target nodes).*

IV.5.3 Cluster linkage between reference and confounded dataset

In order to conduct a joint analysis, it was necessary to standardize the number of pathways for comparison between datasets, resulting in a final count of 50 pathways. The initial step involved identifying flow patterns between the two technical repetitions of the balanced study, followed by comparing repetition A with the confounded study. The results were summarized in the form of heatmaps (**Figure 51**).

There is a very similar flow layout observed between the comparisons of repA_vs_repB and repA_vs_confounded (**Figure 52**). When considering the T1 timepoint, a specific cluster from repetition A exhibits similarities to two clusters from either repetition B or the confounded study. However, in most cases, one of the flows is usually stronger, particularly in the repA_vs_repB comparison.

In the T2 and T3 timepoints, one-to-one flows occur. For instance, T2_II_2 from repA only flows to T2_I_2 of repB. However, when compared to the confounded study, it demonstrates similarity to theoretically different clusters: T2_II_1 and T2_II_2.

In the comparison between repA and repB (repA_vs_repB), the cluster with the highest similarity score is T1_II_2, which closely corresponds to the cluster T1_I_2 from the confounded study (**Table 7**). Following that, the subsequent cluster in the repA_vs_repB comparison, T1_I_2, displays the greatest similarity to T1_II_1 of the confounded study.
In repA, T1_I_1 can be associated with T1_II_2 from the confounded study. The last pair in the repA_vs_repB comparison is identical to the last pair in the repA_vs_confounded comparison. For timepoints T2 and T3, the order of clusters from repA in the repA_vs_repB comparison remains the same as the order in the repA_vs_confounded comparison.

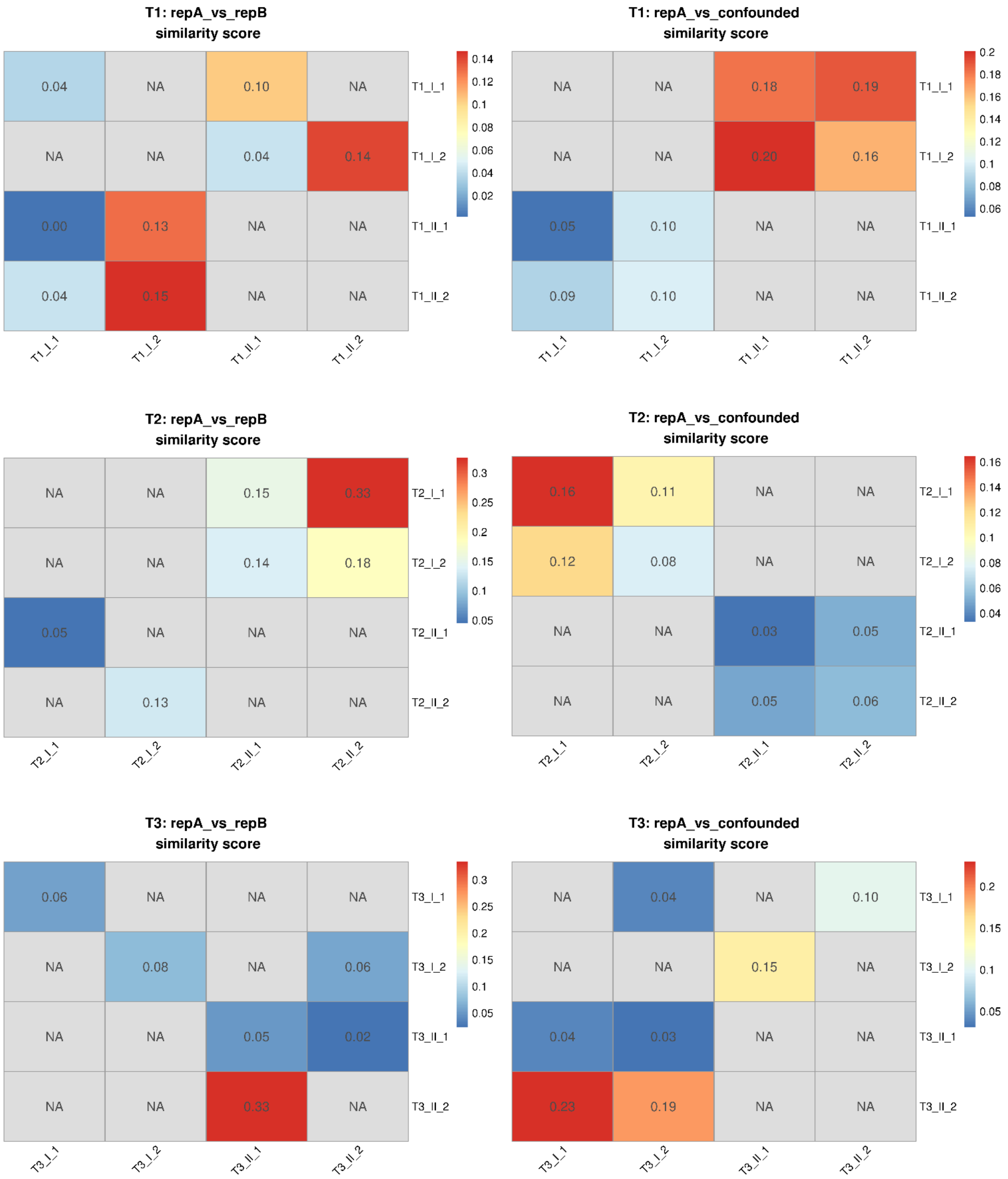


Figure 51. *Heatmaps of the similarity scores between clusters of corresponding timepoints across datasets. The left column represents the comparisons between technical replicates of the balanced study (repA_vs_repB), while the right column corresponds to the comparison of repetition A of the balanced study with the confounded study (repA_vs_confounded). The row names indicate the clusters from repetition A. NA – indicates negative value of the similarity score.*

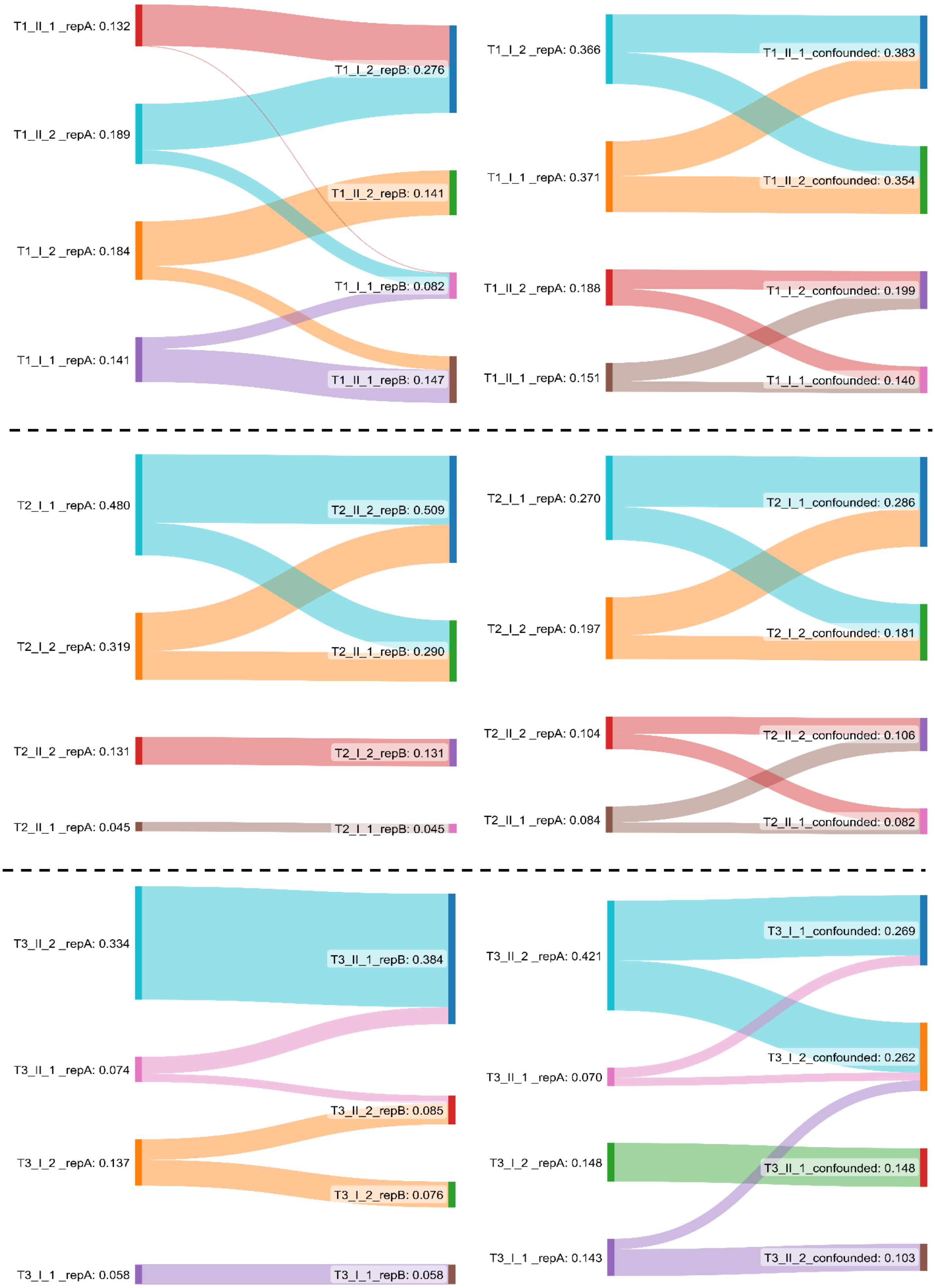
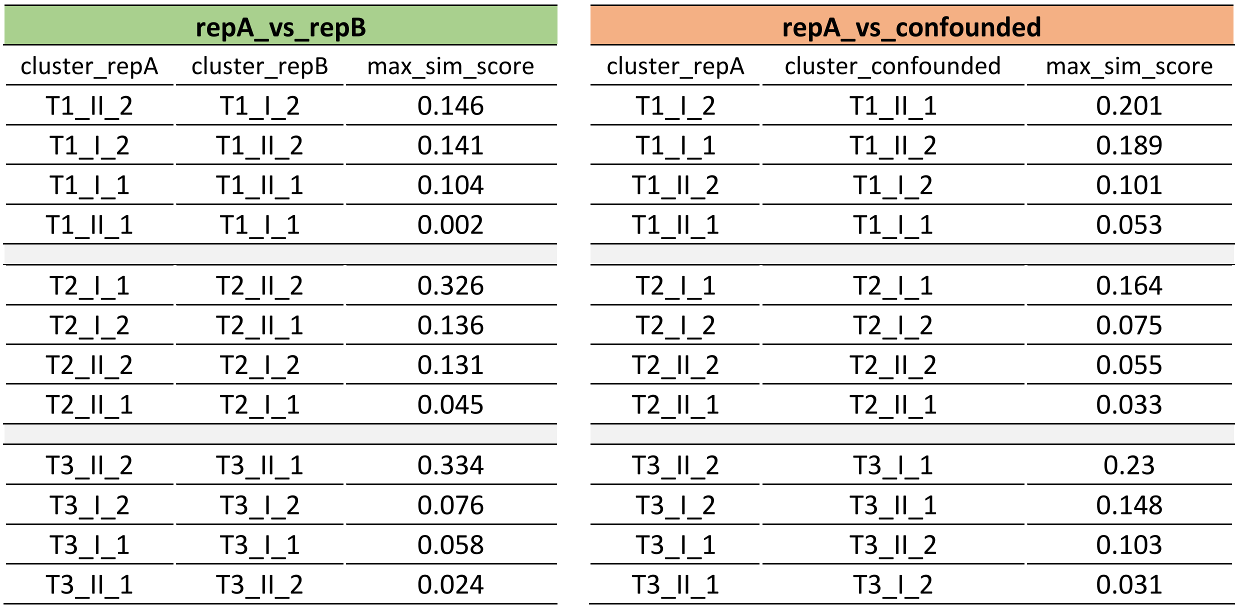


Figure 52. *Sankey diagrams for between datasets comparisons. Left column corresponds to the comparison between repetition A and B of balanced study. Right column corresponds to the comparison between repetition A of balanced study and confounded study. Rows reflect the corresponding timepoints: top – T1; middle – T2 and bottom – T3. The thickness of flows is proportional to the value of similarity score. The label on the right of the cluster name represents the sum of the values coming out of the given cluster (for source nodes) or the sum of the values coming in the given cluster (for target nodes).*

Table 7. *Pairs of clusters with the maximum values of similarity score between datasets*



repA_vs_repB			repA_vs_confounded		
cluster_repA	cluster_repB	max_sim_score	cluster_repA	cluster_confounded	max_sim_score
T1_II_2	T1_I_2	0.146	T1_I_2	T1_II_1	0.201
T1_I_2	T1_II_2	0.141	T1_I_1	T1_II_2	0.189
T1_I_1	T1_II_1	0.104	T1_II_2	T1_I_2	0.101
T1_II_1	T1_I_1	0.002	T1_II_1	T1_I_1	0.053
T2_I_1	T2_II_2	0.326	T2_I_1	T2_I_1	0.164
T2_I_2	T2_II_1	0.136	T2_I_2	T2_I_2	0.075
T2_II_2	T2_I_2	0.131	T2_II_2	T2_II_2	0.055
T2_II_1	T2_I_1	0.045	T2_II_1	T2_II_1	0.033
T3_II_2	T3_II_1	0.334	T3_II_2	T3_I_1	0.23
T3_I_2	T3_I_2	0.076	T3_I_2	T3_II_1	0.148
T3_I_1	T3_I_1	0.058	T3_I_1	T3_II_2	0.103
T3_II_1	T3_II_2	0.024	T3_II_1	T3_I_2	0.031

V. DISCUSSION

Batch effects pose an inevitable challenge in large-scale experiments and those involving multiple omics layers. They introduce an additional layer of technical noise to an already noisy scRNAseq data. However, this noise is not uniformly distributed across genomic data features [87], making it unsuitable to address during the normalization step. Consequently, computational correction or removal becomes necessary, which is the objective of existing algorithms. Nevertheless, distinguishing batch effects from biological heterogeneity is a challenging task due to their differential origins.

Although batch effects have a detrimental impact on the data, the process of correction for them can also be harmful, particularly at the gene-level. There are several downsides to batch correction, including the lack of a measure to quantify the uncertainty associated with the correction process, requiring caution in the application of correction tools. Existing algorithms often prioritize achieving complete mixing of cells between batches rather than preserving the underlying population structure. Furthermore, it was shown in this work that batch correction distorts the original data distribution, making gene-level analyses infeasible. Additionally, there is no single best method that can be applied to all datasets and experimental setups.

The advancement of this field of data analysis primarily focuses on developing new tools that overcome the limitations of previous approaches. However, it is important to note that computationally correcting completely confounded experiments is infeasible. In this PhD project, an attempt was made to tackle the challenge of analyzing completely confounded datasets without the need for batch correction.

To facilitate the consolidated analysis of separately generated data, a pipeline utilizing iterative subspace clustering was proposed. Several publications have demonstrated the usefulness of subspace clustering in mitigating noise in scRNA-seq data [137-139]. However, an approach to utilize this technique specifically for mitigating batch effects has not been explored yet. Furthermore, subspace clustering framework was combined with functional analysis of gene pathways. The novel and central idea was to employ effect size measure to determine cluster-specific pathways, followed by a linkage procedure that enables cluster tracking across different batches. To address specific challenges encountered in scRNA-seq data, original adjustments were introduced, such as global noise filtration based on binarized gene expression matrix to handle dropouts. This project stands out due to its unique experimental setup, which involves a pair of experimentally derived datasets. These datasets shared identical biological properties but differed only in technical aspects. This setup is distinct from existing evaluations of batch effect correction, which often rely on simulation scenarios or include true cell identity labels. Simulation studies are appealing because they allow for the definition of a ground truth, which is often challenging to establish in experimental data. However, simulations cannot fully replicate real-life experimental data and may introduce artificial effects [140, 141]. Therefore, the utilization of a real-life evaluation setup provided ideal conditions for exploration.

The balanced dataset, consisting of technical repetitions labeled as A and B, served as a two-level validation set. In the initial step, this dataset was used to validate the feature selection strategy based on decomposing gene variances into mixtures of Gaussian components. The results demonstrated that the sparse k-means algorithm, coupled with a more sophisticated feature selection strategy, assigned higher importance to highly variable genes identified by the Gaussian Mixture Model (GMM) strategy. Furthermore, the performance of clustering, based on only a small fraction of highly informative genes, showed significant improvement. However, even after discarding the majority of noisy genes during the global filtration step, the signal-to-noise ratio remained insufficient. This might have resulted in some highly variable genes being given low importance. The low signal-to-noise ratio also contributed to the high overlap of Gaussian components. Additionally, near-zero counts observed in the expression matrix had a significant impact on the accuracy of variance calculation, as the squared differences from the mean became even smaller, leading to numerical instability.

The Cliff’s delta effect size statistics is widely recognized. However, its application for determining cluster-specific pathways has not been previously investigated. The evaluation scenario, known as "one-to-others," was designed to identify pathways with a robust manifestation. These pathways were assumed to demonstrate resilience to the negative impact of batch effects, which is generally lower compared to individual genes. In the functional analysis step, a collection of 186 KEGG pathways was utilized.

Although, not all of the KEGG pathways were relevant for the analysis of cancer cells, the linkage between clusters was feasible.

In the initial step, cluster tracking was conducted between batches corresponding to different timepoints. This approach was motivated by the understanding that not all cells respond equally to drug administration, and there may be unaffected cell clusters in subsequent timepoints. In other words, certain cancer cells may develop resistance to the drug, while others may not. The analysis revealed intricate flow patterns between the technical repetitions of the balanced study. It was observed that a specific cluster could exhibit similarities with multiple other clusters, which is expected due to the inherent cellular heterogeneity even within a homogenous cell culture.

It is important to note that k-means clustering employs a random initialization procedure, which means that the initial cluster centers are selected stochastically. Consequently, different cluster assignments may result in the final outcome. Therefore, the cluster labelled as T2_II_1 in one dataset may not differ significantly from the cluster labeled as T2_II_2 in the second dataset, as the final labels were assigned within the same splitting depth.

The second scenario involved establishing a linkage between corresponding timepoints of a reference and a confounded study. In this scenario, the flow patterns discovered in the reference dataset were transferred to the confounded dataset, revealing more uniform flow structures. The values of similarity scores may appear low or insignificant. However, it should be emphasized that the interpretation of similarity scores does not follow strict rules like a correlation coefficient, and should be adjusted to specific experimental conditions. It is crucial to keep in mind that the analysis focused on a completely confounded dataset, representing the most extreme case. The obtained results suggest that the proposed approach may be a promising avenue of investigation. The presented approach may serve as a last resort protocol for analyzing confounded from scRNAseq experiments.

The proposed workflow prioritizes simplicity, low computational cost, and ease of interpretation. All the methods employed in this pipeline are well-established and widely recognized in the field. However, it is worth noting that the goal, which was not initially introduced in this dissertation, was to provide an approach that is easily understandable not only for statisticians or data analysts but also for biologists responsible for designing such experiments.

The use of hierarchical clustering in the global filtering step was the most computationally demanding step of our proposed pipeline, and one may question its utility. However, this type of clustering is easy for interpretation and allows for the implementation of various similarity metrics. Dropouts could also be addressed through imputation. However, it's important to note that this strategy can only propagate biases that are already present in the batch effect-affected dataset.

There is potential for improvement in terms of unsupervised splitting as well. K-means clustering is based on Euclidean distance which may not be ideal metric for highly or even moderately sparse data. However, k-means is straightforward to analyze and interpret the clustering results, as it assigns data points to clusters based on their proximity to the cluster centers, making it intuitive and simple to implement. In contrast, other algorithms like graph-based methods may generate more abstract cluster representations, which can make result interpretation less straightforward. In this study, two rounds of clustering were conducted, which proved sufficient for enabling cluster linkage. However, further explorations involving greater depth should be undertaken. Additionally, it is important to explore alternative pathway collections or consider creating a customized collection that is specifically relevant to the biological system under investigation.

To fully validate the proposed approach, further research is necessary, addressing the aforementioned issues. Nonetheless, it is important to note that this work aimed not to provide a ready-to-use method but rather to pave the way for new directions in research.

SUMMARY

The results presented in this dissertation justify the theses presented in Section I.2, particularly:

- **Section IV.1** proves that batch correction have a detrimental effect on the original distribution of scRNAseq data, making gene-level analysis infeasible.

- **Section IV.4** proves that utilizing 2-level filtration strategy improves the performance of clustering. Furthermore, simple feature selection method based on variance decomposition can substitute more sophisticated and black-box algorithms.
- **Section IV.5** proves that combining functional analysis and cluster linkage procedure allows to skip batch correction step.

ABSTRACT

Single-cell RNAseq experiments are often conducted on a large scale, involving multiple laboratories or measurements taken at different times. Perfectly balanced experimental designs for such large projects may be infeasible, resulting in the need to conduct experiments in batches. Consequently, batch effects inevitably arise. Batch effects introduce variation that is unrelated to the biological variability under investigation, thereby obscuring it. If left unaddressed, batch effects can result in misleading conclusions drawn from the analysis. Therefore, batch effects have to be computationally corrected or removed.

Although batch effects have a detrimental impact on the data, the process of correction for them can also be harmful, particularly at the gene-level. There are several downsides to batch correction, including the lack of a measure to quantify the uncertainty associated with the correction process, requiring caution in the application of correction tools. Existing algorithms often prioritize achieving complete mixing of cells between batches rather than preserving the underlying population structure.

This work aims to provide a pipeline that utilizes iterative subspace clustering, combined with functional analysis of gene sets, to mitigate the negative impact of the batch effect on scRNAseq data. The crucial aspect of the functional analysis involves identifying cluster-specific pathways and establishing their linkage between batches. Therefore, the proposed workflow eliminates the need for applying batch-effect correction and enables consolidated analysis of batches that were generated separately.

STRESZCZENIE

Efekt paczki jest nieuniknionym zjawiskiem w przypadku wysokoprzepustowych i wielkoskalowych eksperymentów, gdzie ograniczenia logistyczne wymagają generowania danych w różnym czasie i przy zaangażowaniu wielu laboratoriów, często wyposażonych w odmienne platformy sprzętowe, wykorzystujących różne partie odczynników i przy udziale zróżnicowanego personelu badawczego.

Wspólna analiza takich danych jest niewykonalna, ponieważ efekty paczki przesłaniają badaną zmienność biologiczną. Takie dane należy skorygować, aby konkluzje wyciągnięte z ich analizy były wiarygodne. Niestety sam proces korekty wiąże się z kilkoma negatywnymi konsekwencjami. Korekta zniekształca bowiem pierwotną naturę oraz dystrybucję danych. Ponadto brakuje miary do ilościowego szacowania niepewności tego procesu. Co więcej, korekta z wykorzystaniem narzędzi bioinformatycznych jest praktycznie niemożliwa w przypadkach, gdzie badana zmienna biologiczna jest całkowicie skorelowana ze zmienną techniczną.

W niniejszej pracy zaproponowano podejście umożliwiające skonsolidowaną analizę zestawów danych pochodzących z eksperymentów scRNAseq i prezentujących. silny efekt paczki. Proponowane rozwiązanie opiera się na iteracyjnej metodzie grupowania z selekcją cech połączonej z analizą funkcjonalną zestawów genów. Po analizie funkcjonalnej otrzymane klastry są łączone między paczkami na podstawie ich funkcjonalnego podobieństwa. Celem takiego podejścia jest złagodzenie negatywnego wpływu efektu paczki bez konieczności jego korekty i związanymi z tym konsekwencjami.

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my supervisor, **Prof. Joanna Polańska**, for her invaluable guidance and unwavering support throughout this project, and for believing in my abilities.

I am also thankful to **Dr Michał Marczyk** for his patience and willingness to answer my numerous questions, no matter how trivial. His guidance has not only enhanced my understanding of the subject matter but has also fostered my intellectual growth.

I am thankful to my **wife Katarzyna** for the big sacrifices she made to ensure that I had the time and space to focus on my work. Her understanding and willingness to adapt to the demands of this project have been remarkable.

This study has been financially supported by AIDA grant:
POWR.03.02.00–00-I029.
Computations were carried out using the GeCONiI infrastructure.

VI. BIBLIOGRAPHY

1. Haque, A., et al., *A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications.* Genome Med, 2017. **9**(1): p. 75.
2. Sun, Y.V. and Y.J. Hu, *Integrative Analysis of Multi-omics Data for Discovery and Functional Studies of Complex Human Diseases.* Adv Genet, 2016. **93**: p. 147-90.
3. He, D. and L. Xie, *A cross-level information transmission network for hierarchical omics data integration and phenotype prediction from a new genotype.* Bioinformatics, 2021. **38**(1): p. 204-210.
4. Hwang, B., J.H. Lee, and D. Bang, *Single-cell RNA sequencing technologies and bioinformatics pipelines.* Exp Mol Med, 2018. **50**(8): p. 1-14.
5. Shalek, A.K., et al., *Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells.* Nature, 2013. **498**(7453): p. 236-40.
6. Vinuela, A., et al., *Age-dependent changes in mean and variance of gene expression across tissues in a twin cohort.* Hum Mol Genet, 2018. **27**(4): p. 732-741.
7. Tomancak, P., et al., *Global analysis of patterns of gene expression during Drosophila embryogenesis.* Genome Biol, 2007. **8**(7): p. R145.
8. Murray, J.I., et al., *Diverse and specific gene expression responses to stresses in cultured human cells.* Mol Biol Cell, 2004. **15**(5): p. 2361-74.
9. Griffiths, J.A., A. Scialdone, and J.C. Marioni, *Using single-cell genomics to understand developmental processes and cell fate decisions.* Mol Syst Biol, 2018. **14**(4): p. e8046.
10. Birnbaum, K.D., *Power in Numbers: Single-Cell RNA-Seq Strategies to Dissect Complex Tissues.* Annu Rev Genet, 2018. **52**: p. 203-221.
11. Wang, D. and S. Bodovitz, *Single cell analysis: the new frontier in 'omics'.* Trends Biotechnol, 2010. **28**(6): p. 281-90.
12. Tang, F., et al., *mRNA-Seq whole-transcriptome analysis of a single cell.* Nat Methods, 2009. **6**(5): p. 377-82.
13. Clark, S. *Single cell RNA-seq: An introductory overview and tools for getting started*. 2022 [cited 2023 03-2023]; Available from: https://www.10xgenomics.com/blog/single-cell-rna-seq-an-introductory-overview-and-tools-for-getting-started.
14. Han, X., et al., *Construction of a human cell landscape at single-cell level.* Nature, 2020. **581**(7808): p. 303-309.
15. Farrell, J.A., et al., *Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis.* Science, 2018. **360**(6392).
16. Zeisel, A., et al., *Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq.* Science, 2015. **347**(6226): p. 1138-42.
17. Navin, N., et al., *Tumour evolution inferred by single-cell sequencing.* Nature, 2011. **472**(7341): p. 90-4.
18. Tirosh, I., et al., *Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq.* Science, 2016. **352**(6282): p. 189-96.
19. *Method of the year 2013.* Nat Methods, 2014. **11**(1): p. 1.
20. *Method of the Year 2019: Single-cell multimodal omics.* Nat Methods, 2020. **17**(1): p. 1.
21. Liu, S. and C. Trapnell, *Single-cell transcriptome sequencing: recent advances and remaining challenges [version 1; peer review: 2 approved].* F1000Research, 2016. **5**(182).
22. Cochain, C., et al., *Single-Cell RNA-Seq Reveals the Transcriptional Landscape and Heterogeneity of Aortic Macrophages in Murine Atherosclerosis.* Circ Res, 2018. **122**(12): p. 1661-1674.

23. Patel, A.P., et al., *Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma.* Science, 2014. **344**(6190): p. 1396-401.

24. Montoro, D.T., et al., *A revised airway epithelial hierarchy includes CFTR-expressing ionocytes.* Nature, 2018. **560**(7718): p. 319-324.

25. Bischoff, P., et al., *Single-cell RNA sequencing reveals distinct tumor microenvironmental patterns in lung adenocarcinoma.* Oncogene, 2021. **40**(50): p. 6748-6758.

26. Kim, J., Z. Xu, and P.A. Marignani, *Single-cell RNA sequencing for the identification of early-stage lung cancer biomarkers from circulating blood.* NPJ Genom Med, 2021. **6**(1): p. 87.

27. Wang, L., et al., *Single-cell RNA-seq reveals the immune escape and drug resistance mechanisms of mantle cell lymphoma.* Cancer Biol Med, 2020. **17**(3): p. 726-739.

28. Patwardhan, G.A., et al., *Treatment scheduling effects on the evolution of drug resistance in heterogeneous cancer cell populations.* NPJ Breast Cancer, 2021. **7**(1): p. 60.

29. Regev, A., et al., *The Human Cell Atlas.* Elife, 2017. **6**.

30. Tabula Muris, C., et al., *Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris.* Nature, 2018. **562**(7727): p. 367-372.

31. Li, H., et al., *Fly Cell Atlas: A single-nucleus transcriptomic atlas of the adult fruit fly.* Science, 2022. **375**(6584): p. eabk2432.

32. Gupta, A., et al., *Inferring gene regulation from stochastic transcriptional variation across single cells at steady state.* Proc Natl Acad Sci U S A, 2022. **119**(34): p. e2207392119.

33. Suter, D.M., et al., *Mammalian genes are transcribed with widely different bursting kinetics.* Science, 2011. **332**(6028): p. 472-4.

34. Raj, A. and A. van Oudenaarden, *Nature, nurture, or chance: stochastic gene expression and its consequences.* Cell, 2008. **135**(2): p. 216-26.

35. Shahrezaei, V. and P.S. Swain, *Analytical distributions for stochastic gene expression.* Proc Natl Acad Sci U S A, 2008. **105**(45): p. 17256-61.

36. Kim, J.K. and J.C. Marioni, *Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data.* Genome Biol, 2013. **14**(1): p. R7.

37. Ochiai, H., et al., *Genome-wide kinetic properties of transcriptional bursting in mouse embryonic stem cells.* Sci Adv, 2020. **6**(25): p. eaaz6699.

38. Jackson, C.A., et al., *Gene regulatory network reconstruction using single-cell RNA sequencing of barcoded genotypes in diverse environments.* Elife, 2020. **9**.

39. Padovan-Merhar, O. and A. Raj, *Using variability in gene expression as a tool for studying gene regulation.* Wiley Interdiscip Rev Syst Biol Med, 2013. **5**(6): p. 751-9.

40. Gross, A., et al., *Technologies for Single-Cell Isolation.* Int J Mol Sci, 2015. **16**(8): p. 16897-919.

41. Kolodziejczyk, A.A., et al., *The technology and biology of single-cell RNA sequencing.* Mol Cell, 2015. **58**(4): p. 610-20.

42. Kim, D., K.B. Chung, and T.G. Kim, *Application of single-cell RNA sequencing on human skin: Technical evolution and challenges.* J Dermatol Sci, 2020. **99**(2): p. 74-81.

43. van den Brink, S.C., et al., *Single-cell sequencing reveals dissociation-induced gene expression in tissue subpopulations.* Nat Methods, 2017. **14**(10): p. 935-936.

44. Adam, M., A.S. Potter, and S.S. Potter, *Psychrophilic proteases dramatically reduce single-cell RNA-seq artifacts: a molecular atlas of kidney development.* Development, 2017. **144**(19): p. 3625-3632.

45. Nguyen, A., et al., *Single Cell RNA Sequencing of Rare Immune Cell Populations.* Front Immunol, 2018. **9**: p. 1553.

46. Adil, A., et al., *Single-Cell Transcriptomics: Current Methods and Challenges in Data Acquisition and Analysis.* Frontiers in neuroscience, 2021. **15**: p. 591122-591122.

47. McDowell, D.G., N.A. Burns, and H.C. Parkes, *Localised sequence regions possessing high melting temperatures prevent the amplification of a DNA mimic in competitive PCR.* Nucleic Acids Res, 1998. **26**(14): p. 3340-7.
48. Jovic, D., et al., *Single-cell RNA sequencing technologies and applications: A brief overview.* Clin Transl Med, 2022. **12**(3): p. e694.
49. Dal Molin, A. and B. Di Camillo, *How to design a single-cell RNA-sequencing experiment: pitfalls, challenges and perspectives.* Brief Bioinform, 2019. **20**(4): p. 1384-1394.
50. de Klerk, E., J.T. den Dunnen, and P.A. t Hoen, *RNA sequencing: from tag-based profiling to resolving complete transcript structure.* Cell Mol Life Sci, 2014. **71**(18): p. 3537-51.
51. Oshlack, A. and M.J. Wakefield, *Transcript length bias in RNA-seq data confounds systems biology.* Biol Direct, 2009. **4**: p. 14.
52. Phipson, B., L. Zappia, and A. Oshlack, *Gene length and detection bias in single cell RNA sequencing protocols.* F1000Res, 2017. **6**: p. 595.
53. Chen, X., S.A. Teichmann, and K.B. Meyer, *From Tissues to Cell Types and Back: Single-Cell Gene Expression Analysis of Tissue Architecture.* Annual Review of Biomedical Data Science, 2018. **1**(1): p. 29-51.
54. Jaitin, D.A., et al., *Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types.* Science, 2014. **343**(6172): p. 776-9.
55. Picelli, S., et al., *Smart-seq2 for sensitive full-length transcriptome profiling in single cells.* Nat Methods, 2013. **10**(11): p. 1096-8.
56. Zheng, G.X., et al., *Massively parallel digital transcriptional profiling of single cells.* Nat Commun, 2017. **8**: p. 14049.
57. Ding, J., et al., *Systematic comparison of single-cell and single-nucleus RNA-sequencing methods.* Nat Biotechnol, 2020. **38**(6): p. 737-746.
58. Ziegenhain, C., et al., *Comparative Analysis of Single-Cell RNA Sequencing Methods.* Mol Cell, 2017. **65**(4): p. 631-643 e4.
59. Svensson, V., et al., *Power analysis of single-cell RNA-sequencing experiments.* Nat Methods, 2017. **14**(4): p. 381-387.
60. Wagner, A., A. Regev, and N. Yosef, *Revealing the vectors of cellular identity with single-cell genomics.* Nat Biotechnol, 2016. **34**(11): p. 1145-1160.
61. Hafemeister, C. and R. Satija, *Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression.* Genome Biol, 2019. **20**(1): p. 296.
62. Bacher, R., et al., *SCnorm: robust normalization of single-cell RNA-seq data.* Nat Methods, 2017. **14**(6): p. 584-586.
63. Lun, A.T., K. Bach, and J.C. Marioni, *Pooling across cells to normalize single-cell RNA sequencing data with many zero counts.* Genome Biol, 2016. **17**: p. 75.
64. Vallejos, C.A., et al., *Normalizing single-cell RNA sequencing data: challenges and opportunities.* Nat Methods, 2017. **14**(6): p. 565-571.
65. Jiang, R., et al., *Statistics or biology: the zero-inflation controversy about scRNA-seq data.* Genome Biol, 2022. **23**(1): p. 31.
66. Sarkar, A. and M. Stephens, *Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis.* Nat Genet, 2021. **53**(6): p. 770-777.
67. Andrews, T.S. and M. Hemberg, *M3Drop: dropout-based feature selection for scRNASeq.* Bioinformatics, 2019. **35**(16): p. 2865-2867.
68. Hicks, S.C., et al., *Missing data and technical variability in single-cell RNA-sequencing experiments.* Biostatistics, 2018. **19**(4): p. 562-578.
69. Pierson, E. and C. Yau, *ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis.* Genome Biol, 2015. **16**: p. 241.
70. Lin, P., M. Troup, and J.W. Ho, *CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data.* Genome Biol, 2017. **18**(1): p. 59.

71. Choi, K., et al., *Bayesian model selection reveals biological origins of zero inflation in single-cell transcriptomics.* Genome Biol, 2020. **21**(1): p. 183.

72. Kharchenko, P.V., L. Silberstein, and D.T. Scadden, *Bayesian approach to single-cell differential expression analysis.* Nat Methods, 2014. **11**(7): p. 740-2.

73. Finak, G., et al., *MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data.* Genome Biol, 2015. **16**: p. 278.

74. Svensson, V., *Droplet scRNA-seq is not zero-inflated.* Nat Biotechnol, 2020. **38**(2): p. 147-150.

75. van Dijk, D., et al., *Recovering Gene Interactions from Single-Cell Data Using Data Diffusion.* Cell, 2018. **174**(3): p. 716-729 e27.

76. Gong, W., et al., *DrImpute: imputing dropout events in single cell RNA sequencing data.* BMC Bioinformatics, 2018. **19**(1): p. 220.

77. Huang, M., et al., *SAVER: gene expression recovery for single-cell RNA sequencing.* Nat Methods, 2018. **15**(7): p. 539-542.

78. Arisdakessian, C., et al., *DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data.* Genome Biol, 2019. **20**(1): p. 211.

79. Li, W.V. and J.J. Li, *An accurate and robust imputation method scImpute for single-cell RNA-seq data.* Nat Commun, 2018. **9**(1): p. 997.

80. Hou, W., et al., *A systematic evaluation of single-cell RNA-sequencing imputation methods.* Genome Biol, 2020. **21**(1): p. 218.

81. Andrews, T. and M. Hemberg, *False signals induced by single-cell imputation [version 2; peer review: 4 approved].* F1000Research, 2019. **7**(1740).

82. Qiu, P., *Embracing the dropouts in single-cell RNA-seq analysis.* Nat Commun, 2020. **11**(1): p. 1169.

83. Kim, T.H., X. Zhou, and M. Chen, *Demystifying "drop-outs" in single-cell UMI data.* Genome Biol, 2020. **21**(1): p. 196.

84. Hicks, S.C., M. Teng, and R.A. Irizarry, *On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data.* bioRxiv, 2015: p. 025528.

85. Zhang, Y., G. Parmigiani, and W.E. Johnson, *ComBat-seq: batch effect adjustment for RNA-seq count data.* NAR Genom Bioinform, 2020. **2**(3): p. lqaa078.

86. Leek, J.T., et al., *The sva package for removing batch effects and other unwanted variation in high-throughput experiments.* Bioinformatics, 2012. **28**(6): p. 882-3.

87. Leek, J.T., et al., *Tackling the widespread and critical impact of batch effects in high-throughput data.* Nat Rev Genet, 2010. **11**(10): p. 733-9.

88. Tran, H.T.N., et al., *A benchmark of batch-effect correction methods for single-cell RNA sequencing data.* Genome Biol, 2020. **21**(1): p. 12.

89. Haghverdi, L., et al., *Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors.* Nat Biotechnol, 2018. **36**(5): p. 421-427.

90. Hie, B., B. Bryson, and B. Berger, *Efficient integration of heterogeneous single-cell transcriptomes using Scanorama.* Nat Biotechnol, 2019. **37**(6): p. 685-691.

91. Lin, Y., et al., *scMerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell RNA-seq datasets.* Proc Natl Acad Sci U S A, 2019. **116**(20): p. 9775-9784.

92. Korsunsky, I., et al., *Fast, sensitive and accurate integration of single-cell data with Harmony.* Nat Methods, 2019. **16**(12): p. 1289-1296.

93. Kujawa, T., M. Marczyk, and J. Polanska, *Influence of single-cell RNA sequencing data integration on the performance of differential gene expression analysis.* Front Genet, 2022. **13**: p. 1009316.

94. Luecken, M.D., et al., *Benchmarking atlas-level data integration in single-cell genomics.* Nature Methods, 2022. **19**(1): p. 41-50.

95. Luecken, M.D. and F.J. Theis, *Current best practices in single-cell RNA-seq analysis: a tutorial.* Mol Syst Biol, 2019. **15**(6): p. e8746.
96. Parekh, S., et al., *zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs.* Gigascience, 2018. **7**(6).
97. Smith, T., A. Heger, and I. Sudbery, *UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy.* Genome Res, 2017. **27**(3): p. 491-499.
98. Dobin, A., et al., *STAR: ultrafast universal RNA-seq aligner.* Bioinformatics, 2013. **29**(1): p. 15-21.
99. Lun, A.T.L., et al., *EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data.* Genome Biol, 2019. **20**(1): p. 63.
100. Tian, L., et al., *scPipe: A flexible R/Bioconductor preprocessing pipeline for single-cell RNA-sequencing data.* PLoS Comput Biol, 2018. **14**(8): p. e1006361.
101. Evans, C., J. Hardin, and D.M. Stoebel, *Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions.* Brief Bioinform, 2018. **19**(5): p. 776-792.
102. Li, B., et al., *RNA-Seq gene expression estimation with read mapping uncertainty.* Bioinformatics, 2010. **26**(4): p. 493-500.
103. Anders, S. and W. Huber, *Differential expression analysis for sequence count data.* Genome Biol, 2010. **11**(10): p. R106.
104. Robinson, M.D. and A. Oshlack, *A scaling normalization method for differential expression analysis of RNA-seq data.* Genome Biol, 2010. **11**(3): p. R25.
105. Brown, J., et al., *Normalization by distributional resampling of high throughput single-cell RNA-sequencing data.* Bioinformatics, 2021. **37**(22): p. 4123-8.
106. Crowell, H.L., et al., *muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data.* Nat Commun, 2020. **11**(1): p. 6077.
107. Lun, A.T., D.J. McCarthy, and J.C. Marioni, *A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor.* F1000Res, 2016. **5**: p. 2122.
108. Brennecke, P., et al., *Accounting for technical noise in single-cell RNA-seq experiments.* Nat Methods, 2013. **10**(11): p. 1093-5.
109. Ranjan, B., et al., *DUBStepR is a scalable correlation-based feature selection method for accurately clustering single-cell data.* Nat Commun, 2021. **12**(1): p. 5849.
110. Pearson, K., *LIII. On lines and planes of closest fit to systems of points in space.* The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1901. **2**(11): p. 559-572.
111. Risso, D., et al., *A general and flexible method for signal extraction from single-cell RNA-seq data.* Nat Commun, 2018. **9**(1): p. 284.
112. Van der Maaten, L. and G. Hinton, *Visualizing data using t-SNE.* Journal of machine learning research, 2008. **9**(11).
113. McInnes, L., J. Healy, and J. Melville, *Umap: Uniform manifold approximation and projection for dimension reduction.* arXiv preprint arXiv:1802.03426, 2018.
114. Kobak, D. and P. Berens, *The art of using t-SNE for single-cell transcriptomics.* Nat Commun, 2019. **10**(1): p. 5416.
115. Kim, T., et al., *Impact of similarity metrics on single-cell RNA-seq data clustering.* Brief Bioinform, 2019. **20**(6): p. 2316-2326.
116. Zurauskiene, J. and C. Yau, *pcaReduce: hierarchical clustering of single cell transcriptional profiles.* BMC Bioinformatics, 2016. **17**: p. 140.
117. Chen, Y.-Z. and Y.-C. Lai, *Universal structural estimator and dynamics approximator for complex networks.* 2016.
118. Kiselev, V.Y., et al., *SC3: consensus clustering of single-cell RNA-seq data.* Nat Methods, 2017. **14**(5): p. 483-486.

119. Grün, D., et al., *Single-cell messenger RNA sequencing reveals rare intestinal cell types.* Nature, 2015. **525**(7568): p. 251-255.

120. Cordeiro de Amorim, R. and B. Mirkin, *Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering.* Pattern Recognition, 2012. **45**(3): p. 1061-1075.

121. Zeng, X. and H. Zheng *CS Sparse K-means: An Algorithm for Cluster-Specific Feature Selection in High-Dimensional Clustering*. 2019. arXiv:1909.12384 DOI: 10.48550/arXiv.1909.12384.

122. Witten, D.M. and R. Tibshirani, *A framework for feature selection in clustering.* J Am Stat Assoc, 2010. **105**(490): p. 713-726.

123. Kondo, Y., M. Salibian-Barrera, and R. Zamar, *RSKC: An R Package for a Robust and Sparse K-Means Clustering Algorithm.* Journal of Statistical Software, 2016. **72**(5): p. 1 - 26.

124. Brodinová, Š., et al., *Robust and sparse k-means clustering for high-dimensional data.* Advances in Data Analysis and Classification, 2019. **13**(4): p. 905-932.

125. Levine, J.H., et al., *Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis.* Cell, 2015. **162**(1): p. 184-97.

126. Xu, C. and Z. Su, *Identification of cell types from single-cell transcriptomes using a novel clustering method.* Bioinformatics, 2015. **31**(12): p. 1974-1980.

127. Duò, A., M. Robinson, and C. Soneson, *A systematic performance evaluation of clustering methods for single-cell RNA-seq data [version 2; peer review: 2 approved].* F1000Research, 2018. **7**(1141).

128. Freytag, S., et al., *Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data [version 1; peer review: 1 approved, 2 approved with reservations].* F1000Research, 2018. **7**(1297).

129. Yu, L., et al., *Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data.* Genome Biol, 2022. **23**(1): p. 49.

130. Kiselev, V.Y., T.S. Andrews, and M. Hemberg, *Challenges in unsupervised clustering of single-cell RNA-seq data.* Nat Rev Genet, 2019. **20**(5): p. 273-282.

131. Marczyk, M., et al., *Multi-Omics Investigation of Innate Navitoclax Resistance in Triple-Negative Breast Cancer Cells.* Cancers (Basel), 2020. **12**(9).

132. Stuart, T., et al., *Comprehensive Integration of Single-Cell Data.* Cell, 2019. **177**(7): p. 1888-1902 e21.

133. McInnes, L. and J. Healy, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.* ArXiv, 2018. **abs/1802.03426**.

134. Mrukwa, G. and J. Polanska, *DiviK: divisive intelligent K-means for hands-free unsupervised clustering in big biological data.* BMC Bioinformatics, 2022. **23**(1): p. 538.

135. Hanzelmann, S., R. Castelo, and J. Guinney, *GSVA: gene set variation analysis for microarray and RNA-seq data.* BMC Bioinformatics, 2013. **14**: p. 7.

136. Zappia, L. and A. Oshlack, *Clustering trees: a visualization for evaluating clusterings at multiple resolutions.* Gigascience, 2018. **7**(7).

137. Zheng, R., et al., *An Adaptive Sparse Subspace Clustering for Cell Type Identification.* Front Genet, 2020. **11**: p. 407.

138. Zheng, R., et al., *SinNLRR: a robust subspace clustering method for cell type detection by non-negative and low-rank representation.* Bioinformatics, 2019. **35**(19): p. 3642-3650.

139. Ning, Z., et al., *A clustering method for small scRNA-seq data based on subspace and weighted distance.* PeerJ, 2023. **11**: p. e14706.

140. Crowell, H.L., et al., *The shaky foundations of simulating single-cell RNA sequencing data.* Genome Biol, 2023. **24**(1): p. 62.

141. Cao, Y., P. Yang, and J.Y.H. Yang, *A benchmark study of simulation methods for single-cell RNA sequencing data.* Nat Commun, 2021. **12**(1): p. 6911.

ABBREVIATIONS:

CV^2 - squared coefficient of variation
cDNA - complementary DNA
DEG - differentially expressed gene
DiviK - Divisive Intelligent K-Means clustering
DNA - Deoxyribonucleic Acid
GMM - Gaussian Mixture Models
GRN – gene regulatory network
GSVA - gene set variation analysis
HC - hierarchical clustering
HVG – highly variable gene
IVT - *in vitro* transcription
KEGG - Kyoto Encyclopedia of Genes and Genomes
MNN - mutual nearest neighbor
mRNA – messenger RNA
PCA – principal component analysis
PCR - polymerase chain reaction
repA – repetition A of balanced study
repB - repetition B of balanced study
RNA - Ribonucleic acid
RT – reverse transcription
SBS - sequencing by synthesis
scRNAseq – single cell RNA sequencing
SVD - singular value decomposition
UMAP - Uniform Manifold Approximation and Projection
UMI – unique molecular identifier

LIST OF FIGURES:

LIST OF TABLES: