Models and methods to process single-cell RNA sequencing data for neuroscience

Dissertation

zur Erlangung des Grades eines
Doktors der Naturwissenschaften

der Mathematisch-Naturwissenschaftlichen Fakultät
und
der Medizinischen Fakultät
der Eberhard-Karls-Universität Tübingen

vorgelegt
von

Jan Lause
aus Aachen, Deutschland

2025

Tag der mündlichen Prüfung: 24.01.2025

Dekan der Math.-Nat. Fakultät: Prof. Dr. Thilo Stehle
Dekan der Medizinischen Fakultät: Prof. Dr. Bernd Pichler

1. Berichterstatter: Prof. Dr. Philipp Berens

2. Berichterstatter: Prof. Dr. Manfred Claassen

Prüfungskommission: Prof. Dr. Philipp Berens

Prof. Dr. Manfred Claassen

Prof. Dr. Sven Nahnsen

Prof. Dr. Kay Nieselt

Erklärung / Declaration:

Ich erkläre, dass ich die zur Promotion eingereichte Arbeit mit dem Titel:

„*Models and methods to process single-cell RNA sequencing data for neuroscience*“

selbständig verfasst, nur die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche gekennzeichnet habe. Ich versichere an Eides statt, dass diese Angaben wahr sind und dass ich nichts verschwiegen habe. Mir ist bekannt, dass die falsche Abgabe einer Versicherung an Eides statt mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft wird.

I hereby declare that I have produced the work entitled “Models and methods to process single-cell RNA sequencing data for neuroscience”, submitted for the award of a doctorate, on my own (without external help), have used only the sources and aids indicated and have marked passages included from other works, whether verbatim or in content, as such. I swear upon oath that these statements are true and that I have not concealed anything. I am aware that making a false declaration under oath is punishable by a term of imprisonment of up to three years or by a fine.

Tübingen, den 31.01.2025 .....

Datum / Date Unterschrift /Signature

all these molecules don't make me who I am

Darwin Deez

Contents

Abstract

Implementing brain functions like vision, memory and cognition requires billions of cells of diverse types to work together. Thus, one important goal in neuroscience is to comprehensively map which cell types exist. These cell types differ in factors such as physiology, morphology and location. Those factors are determined by the transcriptome, the entirety of RNA molecules expressed in each cell. Measuring this "RNA fingerprint" for brain cells with single-cell RNA sequencing therefore became a popular route to understand neural systems. In the last years, large transcriptomic atlas datasets were published, containing the gene expression of ten thousands of genes for millions of brain cells.

However, data from single-cell sequencing is subject to strong technical noise, and thus requires special preprocessing: cell-to-cell variability in sequencing, unequal variances between genes of different expression level, and noise from PCR amplification. Currently, computational biologists often address those noise sources with heuristics for normalization and variance stabilization, but these methods have intrinsic limitations and are poorly motivated by theory.

In addition, single-cell data is high-dimensional and therefore hard to visualize. Many practitioners use PCA followed by nonlinear embedding methods like UMAP or t-SNE to reduce single-cell data to two dimensions. This practice has received substantial criticism, as it is impossible to preserve all aspects of the original data, e.g., high-dimensional distances. As a result, the field currently debates if UMAP and t-SNE should be used at all.

In this thesis, we address challenges in both preprocessing and visualization. For preprocessing, we present a model-based strategy to normalize single-cell RNA sequencing data: Null model Pearson residuals. In this approach, we model the expected technical and statistical noise from the data generation process. Consequently, the residuals of this null model will contain the biologically meaningful signal, and can be used for downstream processing. We show that this approach leads to fast, scalable and effec-

tive normalization, and additionally allows for theoretical insights into the data generation process of single-cell RNA sequencing.

For visualization, we investigate the claim that 2D embeddings of single-cell data are generally arbitrary and misleading. We show that this claim is false and misleading itself, as it was based on inadequate and limited metrics of embedding quality. More appropriate metrics quantifying neighborhood and class preservation reveal that while t-SNE and UMAP embeddings of single-cell data do not preserve high-dimensional distances, they can nevertheless provide biologically relevant information.

Finally, we reflect on future directions for the field of single-cell data preprocessing and visualization, sketch out how neuroscience can build on top of the exciting single-cell work from the last decade, and how this might change how we think about brain cell types in the future.

1 Motivation

In the human brain, roughly 100 billion neurons form the biological circuits (Von Bartheld et al., 2016) that create a continuous experience and control behavior. As neuroscientists, it is our task to better understand the mechanisms in this unimaginably complex system. This is only possible if we first identify the functional building blocks the system is made of. However, we are still missing a comprehensive parts list of the brain: It is even unclear which parts — types of cells — exist in the brain, and what these different parts are good for, i.e., what function a certain type of brain cell is serving in its local circuit. Therefore, we first need to obtain such a parts list, for example by sorting brain cells into cell types that can be defined by function, anatomy or molecular genetics (Ecker et al., 2017; Zeng and Sanes, 2017).

To determine the cell type of neurons experimentally, a multitude of high-throughput methods have been developed that can assess the function and shape of many single cells at the same time. Two-photon imaging with special fluorescent indicator dyes can track voltage-, Calcium- or transmitter levels in living neurons on sub-second to millisecond timescales (Lin and Schnitzer, 2016; Kulkarni and Miller, 2017; Hao and Plested, 2022; Villette et al., 2019; Aggarwal et al., 2023; Zhang et al., 2023), and modern multi-electrode probes directly measure electrical responses of neurons with sub-millisecond resolution (Jun et al., 2017; Hong and Lieber, 2019; Steinmetz et al., 2021). Serial block face electron microscopy (Denk and Horstmann, 2004) and automated transmission electron microscopy (Yin et al., 2020) can reconstruct neurons' morphology and connectivity with nanometer precision (Helmstaedter et al., 2013; MICrONS Consortium et al., 2021).

These methods capture neural activity and how neurons are shaped and connected to each other. However, as all cellular processes, neural function and anatomy are eventually determined by the proteins inside each cell (Hyden, 1967; Matus, 1988; Prasad and Alizadeh, 2019). These complex molecules govern the intra- and intercellular signaling

that neuroscientists are interested in: Proteins and peptides form transmembrane channels and transmitter receptors, serve as intracellular signaling molecules, or even as neurotransmitters. How quickly a neuron will be ready to fire an action potential, or if it is likely to form a new synapse with a neighboring cell is directly reflected by its protein contents. Therefore, assessing the molecular composition of neurons provides neuroscientists with another "view" of each cell's state that is complementary to function and anatomy and allows an even finer cell typing.

With new tools for the molecular characterization of neurons, researchers can now routinely profile every step in single cells' protein production: This process starts from the DNA sequence, which can be assessed by single cell DNA sequencing (Gawad et al., 2016). Which parts of the DNA are transcribed into RNA molecules is regulated by the epigenome, which can be measured, e.g., as the DNA's methylation status (Karemaker and Vermeulen, 2018), its open chromatin accessibility (Buenrostro et al., 2015; Klemm et al., 2019) or histone modifications (Rotem et al., 2015). The transcriptome of a cell — all its RNA molecules — can be assessed with single cell RNA sequencing (scRNA-seq, Kolodziejczyk et al. (2015)). Finally, a subset of RNAs, the messenger RNA (mRNA), is translated into the cell's proteins. Directly counting these proteins is possible with with mass spectrometry methods adapted for single-cell processing (Vistain and Tay, 2021; Specht et al., 2021), but this technology is still in its infancy and relatively low-throughput (e.g., $\sim$ 200 cells per day with SCoPE2, Petelski et al. (2021)). Therefore, instead of measuring proteins directly, researchers that seek to identify cells and their types often resort to scRNA-seq measurements as a readily available proxy for protein levels. Although simultaneous measurements of RNA and protein have shown that this proxy is far from perfect (Koussounadis et al., 2015; Darmanis et al., 2016; Stoeckius et al., 2017), scRNA-seq experiments have led to a multitude of discoveries in neuroscience (reviewed in detail in Section 2.1):

For example, from scRNA-seq atlases we now know that the mammalian retina — the first part of the visual system — has more than twice as many cell types than commonly thought in the pre-scRNA-seq era, and many of the previously known types could be validated by scRNA-seq (Masland, 2012; Shekhar and Sanes, 2021). Similar atlas studies of the cortex have revealed how inhibitory and excitatory neurons develop their regional specificity very differently (Zeisel et al., 2015; Tasic et al., 2018). Subsequent studies of whole-brain preparations aimed to provide a comprehensive description of *all* transcriptomes in the mouse nervous system (Yao et al., 2023), describing a taxonomy of thousands of cell types. Building on these foundations, very recent work investigated how this taxonomy changes in human diseases like Alzheimer's, hoping for insights into disease pathways and new treatments (Mathys et al., 2024). Especially these recent studies have presented overwhelmingly large datasets, raising questions on how to make sense of this vast neural diversity.

Handling and analyzing scRNA-seq data is challenging in itself: After sequencing RNA transcripts and mapping them to the genome, scRNA-seq data comes as counts for each gene in each profiled cell — with tens of thousands distinct genes in typical mammalian data, and up to millions of cells in large experiments. These counts are typically sparse (i.e., in any given cell, only a minority of genes expressed and many counts are zero), and often relatively small for most genes, leading to low intrinsic signal-to-noise levels. In addition, technical noise source factors that vary between cells (like sequencing efficiency) further complicate the situation.

To handle this kind of noisy, high-dimensional data, the field has converged on three major best best-practice steps for preprocessing scRNA-seq count data (Luecken and Theis, 2019; Amezquita et al., 2020): First, a set of normalization steps filters out noise from the raw counts. Second, dimensionality reduction with, e.g., PCA removes redundant genes. Finally, the normalized and dimensionality-reduced data is inspected

with a non-linear 2D visualization tools like t-SNE or UMAP (van der Maaten and Hinton, 2008; McInnes et al., 2018; Kobak and Berens, 2019) to check for processing artifacts and to form first scientific hypotheses about the data.

However, this default pipeline is not without problems: some of the most common data normalization practices rely on *ad-hoc* heuristics with little theoretical motivation (Warton, 2018; Lun, 2018; Ahlmann-Eltze and Huber, 2023) and have underperformed in benchmarks (Cole et al., 2019; Tian et al., 2019). Also, the use of 2D embeddings for single-cell data visualization has received substantial criticism, as these embeddings can introduce substantial distortions (Wattenberg et al., 2016; Wang et al., 2023b), and allegedly lead to "arbitrary" shapes (Chari and Pachter, 2023).

This thesis will address both of these problems in the preprocessing of single-cell data. First, we will present two new, theory-based methods for normalizing scRNA-seq data: analytic Pearson residuals for UMI count data and compound Pearson residuals for non-UMI counts. Both methods are derived from a simple data generation model and provide a fast and effective normalization of the data. Second, we will present a re-analysis of the Chari and Pachter (2023) claim that 2D embeddings are "arbitrary" and hence should not be used. We found that this claim is flawed, and offer more nuanced recommendations on how to use 2D embeddings for single-cell data analysis. In all chapters, we apply the methods in questions to heterogeneous neural datasets to demonstrate their performance in real-world applications.

The rest of this thesis is structured as follows: the Background Chapter showcases how single-cell transcriptomics data has enriched neuroscience and changed how we think about the brain (Section 2.1, briefly describe how scRNA-seq data are generated practically (Section 2.2), and review common practices in scRNA data processing and identify shortcomings and open questions (Section 2.3.1. In Chapters 3-4, we present two theory-based preprocessing methods that circumvent short-

comings of commonly used heuristics. Chapter 5 addresses recent criticisms of 2D embeddings, and offers a more nuanced view on their strengths and weaknesses. Finally, the Discussion chapter provides an outlook into the future of single-cell processing methods and their impact on neuroscience.

2 Background

2.1 What did scRNA-seq contribute to Neuroscience?

Before scRNA-seq technologies were established, neuroscientists interested in gene expression had to resort to microarrays or bulk RNA-seq. In both technologies, the total RNA from all cells in a sample is extracted at once. To identify the RNA molecules collected, they are either captured on an microarray (by probe molecules that bind selected RNA sequences of interest), or the RNA is converted to cDNA and then sequenced by next-generation sequencing (for bulk RNA-seq). Both technologies thus measure the average gene expression in a tissue.

Neuroscientists used average gene expression data, e.g., to better understand brain diseases: For example, one microarray study on experimentally induced strokes in rats investigated which genes are involved in neuronal cell death after stroke, and which genes protect neurons against it (Kawahara et al., 2004). Similarly, a study on rat models for multiple sclerosis screened RNA expression data from diseased animals for highly expressed genes and thereby identified the pro-inflammatory signaling molecule osteopontin as a potential target to slow down the progression of the disease (Chabas et al., 2001). These examples highlight the potential of RNA data for clinical neuroscience, but such studies of average gene expression suffer from several limitations: Observed changes in RNA levels can not definitely be attributed to actual changes in gene expression, but are often due to changes in cell type proportions rather then actual up- or downregulation of individual genes in individual cell types (Srinivasan et al., 2016). Also, if one observes a gene with low expression, this gene could either be equally low expressed in all cells in the sample or be not expressed in most cells but highly active in the few cells of a rare cell type—two indistinguishable scenarios with very different implications for the role of that gene. Generally, the regulatory effects in genes that are specific to subpopulations are hard to observe in average gene expression data.

Single-cell RNA-seq (scRNA-seq) addresses these problems by barcoding all RNA from the same cell with a shared RNA tag before sequencing RNA from all cells in parallel (see Section 2.2 for a review of the experimental details). By using the cell barcode, RNA molecule counts can later be attributed to the single cells they originated from. This allowed neuroscientists to acquire transcriptomes of single cells and explore their diversity within and across brain areas. The following subsections will showcase how scRNA-seq studies took stock of the cell types in increasingly complex tissues: We start from a cell type census in the mouse retina, then move on to the more complex areas in mouse cortex and hippocampus, and eventually attain a census of the whole mouse nervous system.

2.1.1 Local brain cell census: Retina

The retina converts the visual world around us into electrical signals, compresses them, and sends them to the brain. Studying how the retina accomplishes this task has been a fruitful field in neuroscience, because the retina itself can be considered a small model of the brain (Dowling, 1987): It is organized in layers, contains most major brain cell classes (like excitatory and inhibitory neurons with a diverse set of neurotransmitters) and solves a complex and interesting information processing task. At the same time, the retina is easy to study as a system: Experimental access is easy, and there is only one input path (light-sensitive photoreceptors), one output path (the optic nerve) and therefore simpler circuits than, e.g., in the cortex. This simpler structure makes it more likely to understand general principles of neural processing when studying the retina (Ames III and Nesbett, 1981). Also, the retina is strongly conserved across vertebrate species in many ways, such that neural principles in the retina are likely to generalize across species (Baden et al., 2020).

The vertebrate retina receives light input at the rod and cone photoreceptors. The signal then travels through a layer of bipolar cells (BCs,

Franke et al. (2017)) and a layer of retinal ganglion cells (RGCs, Baden et al. (2016)). BCs and RGCs are heterogeneous: They consist of many cell types, each of which processes the input signal differently, thereby forming "channels" that process different features. These features are further sharpened by interneurons (horizontal cells and many types of amacrine cells, ACs) before they are eventually sent to the brain via the optic nerve. To understand how these feature channels arise, it is important to decipher the heterogeneity of all the BC-, AC-, and RGC types that are involved and catalog them.

Single-cell RNA sequencing is the perfect tool to build such atlas datasets, as it can quantitatively profile many cells at once. scRNA-seq is also less laborious than morphology reconstruction and less context-dependent than functional measurements (Shekhar and Sanes, 2021). Macosko et al. (2015) were the first to apply scRNA-seq to the whole mouse retina, recovering all major cell classes. However, as they sampled cells without restriction from the retina, their atlas is dominated by rod photoreceptors, the most numerous cell type in the retina. This motivated future studies to use enrichment sampling via FACS[1], i.e., to sequence only cells with certain markers present. For example, Shekhar et al. (2016) enriched BCs and found all 13 previously known BC types and two new ones. Among the newly discovered was the unusual type BC1B: It has a monopolar morphology, which likely led previous studies to misclassify it as amacrine cell (Shekhar and Sanes, 2021). However, the Shekhar et al. (2016) atlas confirmed that transciptomically, it was indeed closely related to BC type BC1A, and showed no similarity to amacrine cells.

Similar studies revealed 46 RGC types (Tran et al., 2019) and 63 AC types (Yan et al., 2020) — substantially more than previously described with functional or morphological methods. These types were usually identifiable with single or at most 2–3 marker genes. Especially for ACs and BCs, the cell types formed a meaningful hierarchy:

[1] Fluorescence-activated cell sorting (FACS) allows to separate cells based on predefined, fluorescent surface markers.

groups of cell types with similar transcriptome mapped to previously known subclasses, i.e., ON and OFF BCs, or GABAergic and glycinergic ACs (Shekhar and Sanes, 2021). This corroborates the hope that a transcriptomic taxonomy of retinal cell types rather extends and refines previously known subclasses (e.g., by dividing them in finer, new types and even subtypes), rather than behaving orthogonal to previous knowledge.

What are these taxonomies useful for? Shekhar and Sanes (2021) lists three applications: First, starting from the cell type atlas datasets introduced above, one can study how these cell types arise during individual development (Clark et al., 2019). Second, one can combine such atlas datasets across species to study the evolutionary convergence and divergence of individual cell types. Hahn et al. (2023) has attempted this and found unexpected homology between the most important RGCs in primate retinae, midget and parasol cells, and a set of RGC types in the mouse retina, the alpha RGCs — a result that might have direct implications on how mouse retina research is translated to primate and human applications. For example, one could now specifically target mouse alpha RGCs if one aims at treatment of human eye diseases that impact midget or parasol RGCs, e.g., in human glaucoma (Tribble et al., 2019).

A third application beyond individual and evolutionary emergence of cell types is to develop a mechanistic understanding of eye diseases by combining retina scRNA-seq atlas data with genome-wide association studies (GWAS) (Shekhar and Sanes, 2021). GWAS yields risk genes for a certain disease from population-wide comparisons of genomic mutations, but does not tell in which cells the risk genes are expressed. scRNA-seq atlases can fill this gap, and were able to reveal that some disease risk genes are expressed in cell types that were not previously linked to the disease. For example, Yi et al. (2021) showed that a risk gene for age-related macula degeneration (AMD, known to affect mainly cells of the retinal pigment epithelium) is also expressed in horizontal cells and Mueller glia, potentially implicating these cell types in still

unknown disease mechanisms of AMD.

In summary, scRNA-seq atlases have significantly advanced the understanding of the retina as a model system of the brain. They mapped out the cell type diversity in the retina, and showed how retinal cell types are connected across development stages and conserved across species. First results on eye diseases raise hopes that this atlas knowledge might eventually contribute to better understanding and treating these diseases.

2.1.2 Local brain cell census: Cortex and hippocampus

In contrast to the retina, the cortex areas of the mammalian brain are much less understood. The hippocampus is crucial for memory and navigation, and has been a major target to study learning and plasticity. The neocortex is involved in most cognitive functions, including processing of sensory inputs, planning motor activity and executive functions like decision making. This makes these areas exciting targets for neuroscientists—and single-cell transcriptomics.

In one of the first larger-scale scRNA-seq studies of the brain, Zeisel et al. (2015) performed a "cell census" in mouse primary somatosensory cortex (S1) and mouse hippocampus area CA1, profiling roughly 3 000 cells in total. The authors first clustered the cells into nine major groups, six non-neuronal and three neuronal. Already these major neuronal groups revealed an interesting pattern: While pyramidal neurons from cortex and hippocampus formed separate groups, interneurons from both brain areas appeared as a single cluster. This could mean that while projection neurons are highly area-specific, the local processing by interneurons can be similar between cortex and hippocampus. Zeisel et al. (2015) then further clustered each group into subclasses, revealing that indeed the majority of interneuron subclasses appear in both cortex and hippocampus. In contrast, the subclasses of the two pyramidal groups from cortex and hippocampus were not only area-specific, but also specific to layers or hippocampal subregions. With all non-neuronal

subclasses included, Zeisel et al. (2015) report a total of 47 subclasses. Importantly, both the major groups and all subclasses could be identified from a unique combinatorial code made of a small set of transcription factor genes[2], that in some cases were directly related to the function of the subclass in question. Thus, Zeisel et al. (2015) suggest such transcription factor codes as potential mechanism of how the adult brain maintains functional differences between cell types.

Tasic et al. (2016) performed a similar scRNA-seq study in mouse primary visual cortex (V1), using a different sampling strategy: They used multiple *Cre* mouse lines that fluorescently mark known neuronal cell classes in V1, allowing the authors to enrich these cells of interest with FACS. This lead to a neuron-biased sample of ca. 1 7000 cortical cells from mouse V1. Tasic et al. (2016) clustered these cells into 49 core clusters: 23 GABAergic, 19 glutamaterigic and seven non-neuronal. In contrast to Zeisel et al. (2015), Tasic et al. (2016) allowed "intermediate" cluster memberships between core clusters, revealing that roughly 15% of all cells might not conform to discrete clusters in gene expression space, but instead occupy the space between them.

Harris et al. (2018) followed up on this idea of continua between cell type clusters: They profiled the transcriptomes of ca. 3 700 GABAergic neurons from mouse hippocampus area CA1 after FACS-enrichment for *Slc32a1*, a gene involved in GABA uptake into synaptic vesicles. The authors then clustered these cells into 49 clusters, most of which were subtypes of the 23 previously known CA1 interneuron types. Interestingly, Harris et al. (2018) showed that while some clusters seemed discretely separated from all others, most of them showed overlap with a small set of closely related clusters. The authors interpret these partly overlapping clusters as dimensions of continuous variation that are tiled

[2]Transcription factors (TFs) are proteins that regulate the transcription of DNA to mRNA. Groups of TFs form complex regulatory networks (Babu et al., 2004) that can turn on and off the production of whole sets of proteins, initiating complex programs like cell division or cell death. Thus, genes that code for TF proteins are very influential to the fate of a cell, and it is not surprising they play a role in cell type differentiation.

by multiple clusters. Further analysis suggested that one such dimension in area CA1 could be the preferred synaptic target of interneurons: Harris et al. (2018) report a latent factor in CA1 transcriptomes that places the interneuron clusters on a continuum that ranges from types targeting other interneurons to types targeting the distal dendrites, the proximal dendrites or somata and axons of pyramidal cells. These continuous differences in the latent factor were driven by a set of genes involved in electrophysiological and structural properties of the cell, implying that the latent factor could indeed be mechanistically meaningful to understand circuits in CA1. The authors conclude that such a continuous axis of biologically meaningful variation provides a valuable complementary perspective on gene expression to a discrete clustering.

Tasic et al. (2018) reached a new order of magnitude in terms of dataset size, by scaling up their previous approach of enriching neurons for scRNA-seq with transgenic mouse lines and FACS (Tasic et al., 2016). In Tasic et al. (2018), they focus on two cortical areas with quite different function: the "fast" primary visual cortex V1 underlying early visual processing, and the "slow" anterior lateral motor cortex (ALM) underlying higher-level functions such as short-term memory, motor planing and decision making. In total, the authors sampled almost 24 000 neurons, and were able to cluster them into 133 clusters. Tasic et al. (2018) report that almost all GABAergic types were present in both V1 and ALM, while almost all glutamatergic types were area-specific and only occurred in either one. This confirmed earlier results that hinted at an area-specific taxonomy for projection neurons, and a more general taxonomy of interneurons shared between areas (Zeisel et al., 2015). As Tasic et al. (2016) did before, Tasic et al. (2018) found a substantial fraction (ca. 11%) of "intermediate" cells between clusters, which is in line with continuous variation between cell types. In addition, the authors observed substantial variation within types, sometimes as strong as the between-cluster variation. This hints at the difficulty of reconciling the discrete and continuous aspects of transcriptomic types in a single

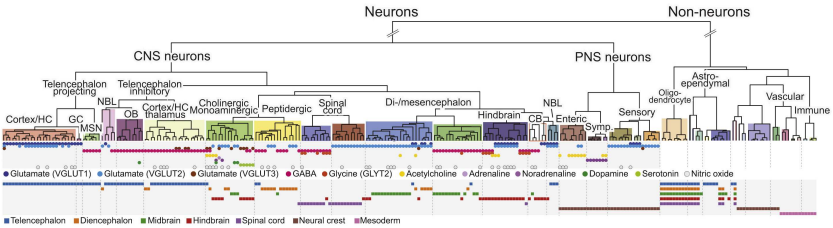

Figure 1: Region-of-origin dominates the transcriptomic taxonomy of the mouse brain neurons. Dendrogram: Hierarchical taxonomy of Zeisel et al. (2018) scRNA-seq brain cell types. Main branches of the taxonomy are separated by colored backgrounds in the dendrogram. Middle panel: Neurotransmitters used per cell type. Lower panel: Developmental origin per cell type. Figure reproduced with modifications from Zeisel et al. (2018) Figure 1c, in compliance with the CC-BY 4.0 license.

framework: Tasic et al. (2018) note that depending on sample size and clustering criteria, types with a lot of continuous within-type variation might be split into discrete subtypes.

In summary, the above selection of scRNA-seq studies of single cortex areas revealed three major patterns: Similar to the retina studies, all studies were able to reconcile their results with previous knowledge about cell types, typically refining existing taxonomies with finer distinctions. Interestingly, excitatory cell types were often area-specific, while inhibitory cell types were shared between areas. Finally, all studies found cases in which cell types were clearly different from each other, but not cleanly separated. Instead, they showed continuous variation in their gene expression, sometimes interpolating between multiple types. This challenges the concept of cell types as discrete entities, and leaves room for theories that explain what function a cell type with an axis of continuous variation might serve in cortex.

2.1.3 Whole-brain cell census

For a more holistic understanding of neural systems, recent studies have attempted to build a comprehensive scRNA atlas of the whole brain or

even all of nervous system. To show the potential impact of such studies, this section will introduce two selected studies in more detail.

Zeisel et al. (2018) were the first to attempt whole-brain scRNA-seq. The authors set out to answer the question which factor mainly drives the molecular fingerprint of cells in the nervous system: the cell's function, its developmental origin or micro-environment. To that end, they sequenced roughly 500 000 cells from all parts of the mouse brain, spinal cord and ganglia of the peripheral nervous system (PNS), omitting only the neural tissue in primary sensory organs like the retina or the inner ear.

As Zeisel et al. (2018) used mostly unbiased sampling, their data were dominated by frequent, non-neural cell classes (e.g., spinal oligodendrocytes made up $> 40\%$ of all cells). Downsampling such over-represented cell types left them with 160 000 cells, which the authors then grouped into 265 hierarchical clusters. This clustering hierarchy suggested that the most important factor driving transcriptomic differences in the nervous system is the difference between neurons and non-neurons (Figure 1). Thousands of genes showed differential expression between those classes, reflecting the specialization of either class, e.g., the ability of neurons to form synapses and axons. Among the neurons, the dominant factor was developmental origin, as neurons from the central nervous system (CNS, developed from the neural tube) and the PNS (developed from the neural crest) separated regardless of functional properties like the type of neurotransmitter used. CNS-neurons split further into distinct groups of cell types for gross anatomical location (and thereby, developmental origin) like telencephalon (including cortex and hippocampus), diencephalon, hind brain and spinal cord, each of which further subdivided into excitatory and inhibitory types. These CNS neuron types were distinguished by genes encoding neurotransmitter production, membrane properties, synaptic structures and region-specific transcription factors, suggesting that these transcriptomic cell types are indeed relevant to understand neural function.

In contrast to the highly area-specific neural cell types, Zeisel et al. (2018) found almost no regional patterning for non-neurons: While CNS oligodendrocytes do originate from the full length along the anterior-posterior axis of the neural tube (like neurons), they seemed to "forget" their origin during development, and did not form region-specific groups. Astrocytes however were the exception: They formed region-specific cell types with very sharp borders that coincided with the borders between neural type regions, suggesting functional coordination between neuron and astrocyte cell types.

In summary, Zeisel et al. (2018) concluded that the major transcriptomic division in the nervous system is between neurons and non-neurons. Neurons and astrocytes then further diversify by developmental origin, while other non-neurons like oligodendrocytes and immune cells do not show such regional patterning. However, the authors note that their clustering is rather conservative, and that due to shallow sequencing (i.e., relatively few RNA molecules sequenced per cell) each of the 265 clusters they described might still contain substantial heterogeneity to be discovered.

Yao et al. (2023) aimed to map out this remaining heterogeneity: As part of the BRAIN Initiative Cell Census Network, they performed microdissections in more than 100 brain areas to collect more than 7 million mouse brain cells for scRNA-seq — a sample size roughly equal to 5% of the total number of cells in a single mouse brain. Their final clustering groups ca. 4 million high-quality cells into seven "neighborhoods", 34 classes, ca. 300 subclasses, ca. 1 200 superclusters and ca. 5 300 clusters.

Afterwards, Yao et al. (2023) selected a subset of 500 genes that optimally distinguished these clusters, and measured their spatial expression patterns in mouse brain slices with MERFISH (Zhang et al., 2021), a spatial transcriptomics technique. They then map each cell in the MERFISH data to its closest cluster in their scRNA-seq data. This allowed to connect the spatial expression patterns from the MERFISH

data to the high-resolution scRNA-seq clusters, providing a fine-grained spatial map for each. This allowed a detailed study of how cell types are distributed in the mouse brain.

The analysis of these spatial maps per cluster revealed that most clusters were specific to rather small sub-regions of the brain, and transcriptomically similar cell types often occupied the same regions. One exception to this were non-neurons, which were present throughout the brain. As previous work found (Zeisel et al., 2015; Tasic et al., 2018), inhibitory interneurons were less area specific than excitatory neurons. Yao et al. (2023) also offered additional insights on the continuous types that previous work reported (Tasic et al., 2016, 2018; Harris et al., 2018): Such intermediate types were more likely to be prevalent across region borders.

Finally, Yao et al. (2023) describe two very different patterns for the evolutionary old and young parts of the mouse brain: The younger pallium with neocortex, hippocampus and thalamus showed cell types that were transcriptomically very different from all other parts of the brain. However, within the Pallium regions, cell types were widely distributed patterns and many of them mixed within subregions. The older regions (midbrain, hindbrain and hypothalamus) showed a very different pattern: In total, there were more cell types in the older parts than in the pallium, especially when normalized for volume (the pallium is much larger). However, these types were extremely specific to their region, and only mixed with a small number of types. This is consistent with the highly specialized nuclei in e.g., the brain stem. Yao et al. (2023) explain these findings with the higher evolutionary pressure on the older parts of the brain (that are related to basic survival functions like feeding, breathing and homeostasis): this pressure prevented these brain parts from diversifying their cell types. In contrast, the younger pallium evolved later and with more tolerance for diversification. Potentially, it is this diversity of cell types that gave rise to the complex cognitive functions that the pallium subserves.

In summary, Zeisel et al. (2018) produced the first whole-brain taxonomy for brain cell types based on scRNA-seq. For neurons, this taxonomy was dominated by developmental origin i.e., types split up in types from CNS vs. PNS, and from Pallium vs. hindbrain. Yao et al. (2023) essentially confirmed these results with an unprecedented sample size and spatial resolution, providing a data resource that lets a comprehensive taxonomy of all brain cell types seem within reach.

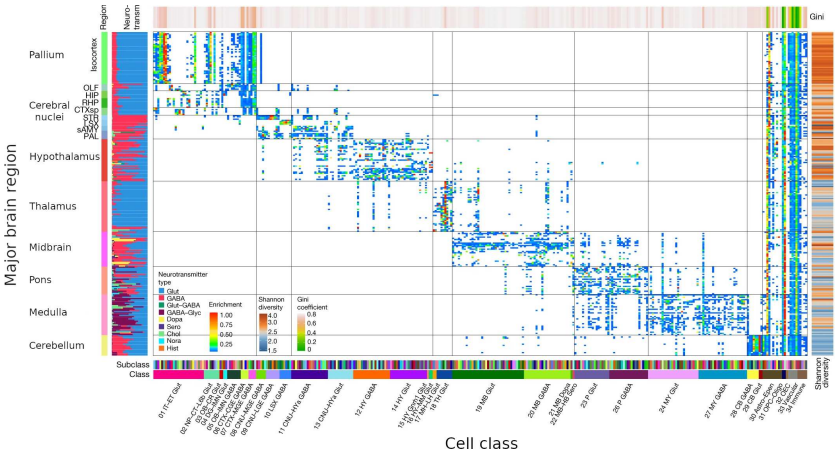

Figure 2: Evolutionary young regions show the highest cell type diversity. Spatial distribution and regional diversity of brain cell types. **Heat map**: Distribution over brain subregions (y-axis) for each of ca. 300 scRNA-seq subclasses (x-axis) from the Yao et al. (2023) BRAIN Cell Census. **Left bar**: Neurotransmitter diversity per subregion. **Bottom bars**: Class and subclass annotations. **Right bar**: *Shannon diversity*, a measure of how complex and diverse the cell type composition is per subregion. Low values (blue) indicate dominance of a few cell types; high values (red) indicate a mixture of many, equally frequent cell types. Note that thalamus, midbrain, pons, medulla and cerebellum tend to show less cell type diversity per subregion than the pallium and cerebral nuclei. **Top bar**: *Gini coefficient*, a measure of how localized a subclass is across regions. High values (red) are very localized, low values (green) very evenly distributed cell types. Note that only non-neuron are spread out across all regions (classes 30–34). Figure reproduced with modifications from Yao et al. (2023) Figure 6a, in compliance with the CC-BY 4.0 license.

2.2 How does single-cell RNA sequencing work?

After the previous section introduced some landmark studies of single-cell transcriptomics in neuroscience, this section will give an overview of how scRNA-seq data is practically generated in the lab.

In scRNA-seq protocols, the goal is to count all RNA molecules in a single cell. That is a challenge for several reasons: First of all, most cells can not be simply sampled in isolation but form tightly connected tissues. Also, once separated, each single cell contains too little amounts of RNA to directly sequence it. Even worse, these RNA molecules are short-lived: the RNA-degrading enzyme RNAse is ubiquitous, and the structure of RNA makes it also chemically unstable. Finally, most experimental protocols for quantification and sequencing of nucleic acid molecules with next-generation sequencing technology are tailored to DNA, not RNA (although there are exceptions like the Nanopore technology (Garalde et al., 2018)).

While individual scRNA-seq techniques might address those challenges in slightly different ways, most high-throughput protocols have converged towards the same main steps: Tissue dissociation, single cell isolation, cell lysis, reverse transcription to cDNA and cell-barcoding, cDNA amplification and finally, cDNA sequencing. This section will outline those steps along with their shortcomings and particularities for the study of neural cells, based on a set of reviews (Kolodziejczyk et al., 2015; Hwang et al., 2018; Tasic, 2018).

Tissue dissociation. First, tissues are dissociated into a cell suspension[3] with the help of proteases and mechanical disruption by repeated pipetting. Depending on the exact conditions, this step can stress cells substantially, leading to altered transcription or even cell death in vulnerable cell types (Kashima et al., 2020). As a result, the cell suspension

[3]In early protocols, cells were collected semi-manually by micromanipulation with micropipettes or laser capture microdissection, sometimes directly from the source tissue. This was laborious and slow, and the high-throughput methods discussed here usually rely on dissociating tissues into a cell suspension.

after dissociation is not perfectly representative of the original tissue. For example, cell suspensions from cortex often lack parvalbumin-expressing cell types (Tasic, 2018), such that these types require special care or artificial enrichment in order to be studied with scRNA-seq. In the special case of neurons, axons and dendrites will be cut off during dissociation along with all the RNA in those compartments. As a result, scRNA-seq of neurons is usually constrained to RNA activity in the nucleus and soma of the cell, missing the RNA that is located at axonal or dendritic translation sites. This "dark transcriptome" can sometimes make up as much as 40% of the whole RNA of a neuron (Cajigas et al., 2012; Ament and Poulopoulos, 2023).

Single-cell isolation. After dissociation, single cells in the suspension need to be captured in separate reaction environments. For that, cells are either sorted into individual wells on a microwell plate, chambers in microfluidic devices or into separate microdroplets of water within a lipid suspension. During sorting, cell types of interest can be enriched by fluorescence-activated cell sorting (FACS), which only allows cells with certain marker proteins enter the single-cell reaction environments.

Cell lysis and reverse transcription. In their isolated reaction environments, each cell is dissolved in a lysis buffer. The RNA in the cell lysate is then reverse-transcribed into cDNA, which is more stable than RNA and allows to use standard PCR and DNA sequencing downstream. However, typical protocols only manage to reverse-transcribe $10 - 20\%$ of all RNA (Kolodziejczyk et al., 2015).

Barcoding. During reverse-transcription, each cDNA sequence that is produced gets tagged with a barcode. This is a short DNA sequence that is unique each reaction environment, and will later allow to tell from the cDNA sequence itself from which single cell each cDNA originated. In addition to this cell-specific barcode, some protocols attach a second type of barcode: a unique molecular identifier (UMI, Islam et al. (2014))

that is unique to every RNA molecule that got reverse-transcribed to a cDNA. That way, UMI barcodes later allow to identify cDNA duplicates that arise during the next step: The amplification phase.

Amplification. After the previous steps, the pool of barcoded and reverse-transcribed cDNA molecules is a too small amount to be sequenced directly. Therefore, non-linear amplification with PCR (polymerase chain reaction) or IVT (in-vitro transcription) is used to copy the existing cDNAs. Both amplification technologies can incur biases: PCR preferentially amplifies certain cDNA sequences, and IVT results in preferential amplification of reads from the 3' end of the original RNA sequence (Kolodziejczyk et al., 2015). If cDNAs were tagged with UMIs in the previous step, this amplification biases can be avoided, while non-UMI protocols suffer from additional amplification noise.

Also, to save resources, some protocols fragment cDNAs and only operate on the 5' or 3' end of the molecules. This is sufficient to later align them to the genome and thereby assign the observed molecule to its gene of origin. But these short snippets will not allow to detect alternative splicing, isoforms or mutations. For this, more costly full-length methods like Smart-seq (Ramsköld et al., 2012) are required.

Sequencing. Finally, the barcoded and amplified cDNA (fragments) from all reaction environments can be pooled and sequenced efficiently with next-generation sequencing (Reis-Filho, 2009). The machinery for sequencing is highly optimized and parallelized, allowing to process large numbers of cDNAs at once. Often, sequencing is done in standardized machines like the Illumina platform.

Sequence alignment. Finally, the observed read sequences are aligned to the genome, thereby mapping transcripts to genes. This step is done by specialized software like STAR (Dobin et al., 2013). In UMI protocols, read duplicates that were generated during amplification are removed based on the UMI tag. After postprocessing, and for the purpose

of basic scRNA-seq experiments, this finally results in the gene expression matrix: It counts for each cell (based on the cell barcode) how many transcripts were found for each gene.

2.3 Which open challenges remain in scRNA-seq data processing?

After reviewing the technical aspects of scRNA-seq data generation, this section will introduce the two major scRNA-seq data processing tasks that this thesis addresses: How can we account for sources of noise in the data (Subsection 2.3.1)? And how can we visualize the high-dimensional data after appropriate preprocessing (Subsection 2.3.2)? Both subsections will introduce the problem setting and existing solutions, before we identify the gaps in current scRNA-seq data processing methods that this thesis will attempt to fill (Subsection 2.3.3).

2.3.1 Preprocessing for noise removal

Most analyses of scRNA-seq data seek to identify which cell types are present in the sample, and which differentially expressed marker genes tell these types apart. From a data science point of view, we start with a count matrix X with cells as rows or observations, and genes as columns or features. Finding cell types means to cluster the rows of the data matrix into groups of cells with similar expression patterns, and finding marker genes is equivalent to select informative columns or features of X that show large variance between the cell clusters. However, finding similar cells and informative genes in scRNA-seq data is hard, as the raw mRNA counts are contaminated by noise. One reason is that each cell only contains small amount of RNA starting material, and depending on the scRNA-seq capture efficiency, only a small fraction of that starting material will eventually be sequenced. This leads to large variability, as Brennecke et al. (2013) showed for artificially high amounts of starting material, even weakly expressed genes could be measured very accurately

with scRNA-seq technology. In contrast, the less starting material was used, the higher mean expression was needed to obtain trustworthy estimates. To deal with this noise in practice, it is helpful to decompose it further into its sources: Sampling noise and sequencing depth noise.

Sampling noise requires variance stabilization. One part of the unwanted variance in scRNA-seq data is intrinsic sampling noise: the observed mRNA counts X can be considered noisy samples from the pool of n mRNAs in a cell, following a binomial distribution

$$X \sim \text{Binom}(n, p_g). \tag{1}$$

Note that the total number of mRNAs in this pool n is large, but the chance that a mRNA from a specific gene is observed p_g is comparatively small. In this situation the law of rare events allows to approximate the Binomial distribution with a Poisson distribution (Lopez-Delisle and Delisle, 2022)

$$X_{\text{Poisson}} \sim \text{Poisson}(\mu = np_g). \tag{2}$$

In practice, the Poisson distribution sometimes underestimates the noise variance (Grün et al., 2014). To account for this additional variance, one can employ the negative binomial (NB) distribution

$$X_{\text{NB}} \sim \text{NB}(\mu = np_g, \theta), \tag{3}$$

where θ is the overdispersion parameter of the NB. Generally, overdispersion refers to additional variance that occurs relative to some model with less variance, in this case the Poisson model. In this parametrization, higher values of θ *decrease* the overdispersion relative to Poisson, and for $\theta = \infty$, there is no more overdispersion and the NB reduces to the Poisson distribution (see Equation 5 below). The negative binomial distribution has already been used to model bulk RNA-seq data (Anders and Huber, 2010) and can be motivated with transcriptional burst-

ing (Raj et al., 2006; Grün et al., 2014), i.e., the observation that a cell's transcription machinery is not constantly producing a certain mRNA species, but instead switches genes on and off in an intrinsically random fashion. This leads to intermittent and irregular periods of very active transcription, contributing additional variance for which the negative binomial distribution can account. Following this interpretation around bursting transcription, some researchers have argued to replace the NB with beta-Poisson model, as its parameters can be directly interpreted as burst size and burst frequency (Wills et al., 2013; Vu et al., 2016). However, most recent work focuses on Poisson and negative binomial models (Hwang et al., 2018).

Poisson and negative binomial counts exhibit heteroscedasticity, i.e., their variance is directly related to the mean

$$\mathrm{Var}[X_{\mathrm{Poisson}}] = \mathbb{E}[X_{\mathrm{Poisson}}] \tag{4}$$

$$\mathrm{Var}[X_{\mathrm{NB}}] = \mathbb{E}[X_{\mathrm{NB}}] + \mathbb{E}[X_{\mathrm{NB}}]^2/\theta. \tag{5}$$

In practice, this implies that genes with a larger mean expression will also have higher variance. This leads to problems in downstream analysis, where we want all informative genes to contribute equally to the analysis task, e.g., the separation of cell types. However, the mean-variance relationship will cause any high-expression gene to have more variance than a low-expression gene—even if the latter happens to be more informative. That is why scRNA-seq preprocessing usually includes *variance stabilization* to remove the mean-variance relation from the raw count data.

Sequencing depth noise requires normalization. Another part of the noise stems from technical factors that introduce differences between cells. Even if one runs scRNA-seq on droplets filled with identical amounts of control mRNA, these artificial cells can differ in their total mRNA counts over orders of magnitude (Grün et al., 2014; Kharchenko, 2021). This is because the biochemistry for capturing, reverse-transcribing

and amplifying mRNA can vary greatly in its efficiency between cells (Stegle et al., 2015; Vallejos et al., 2017; Hwang et al., 2018). For real cells, intrinsic factors like cell size can contribute additional per-cell variation (Stegle et al., 2015). The literature around this topic uses different terms for these factors: Both *sequencing depth* and *library size* refer to the total pool of captured molecules that end up being sequenced for a cell, i.e., its total count across all genes. In this thesis, we call the sum of these per-cell effects *sequencing depth*. Raw mRNA counts are strongly confounded by these cell-to-cell differences in total counts, and the absolute count values cannot be interpreted without *normalization by sequencing depth*.

These two steps, sequencing depth normalization and variance stabilization, are essential for every scRNA-seq preprocessing pipeline[4], and a diverse set of methods has been suggested to perform them. Following the categorization by Ahlmann-Eltze and Huber (2023), the remainder of this section will describe three conceptually distinct classes of preprocessing strategies: (i) global scaling and variance stabilization, (ii) inferring true expression from Bayesian models and (iii) noise model Pearson residuals.

Global scaling and variance stabilization. The most simple strategy combines a global scaling normalization with a non-linear, variance stabilizing transformation (VST) — often the shifted logarithm:

$$X_{cg,\text{normalized}} = \frac{X_{cg}}{s_c} \cdot L \tag{6}$$

$$X_{cg,\text{VST}} = \log(a + X_{cg,\text{normalized}}) \tag{7}$$

[4]Other common tasks include filtering out low-quality cells (e.g., cells that were damaged already before lysis, or droplets/wells that contained multiple or no cells) and the removal of confounding "batch effects", resulting from cells being sampled in separate sequencing rounds (Luecken and Theis, 2019; Amezquita et al., 2020). These tasks are important but beyond the scope of this thesis, and therefore not reviewed in depth here.

where the raw counts X_{cg} per cell c and gene g are normalized by *size factors* s_c, a per-cell measure of sequencing depth (often the total count per cell (Klein et al., 2015; Luecken and Theis, 2019; Amezquita et al., 2020)). L is the total count per cell after normalization, and a is the shifting parameter or *pseudocount* of the logarithm, often set to $a = 1$.

Using the log-transform after global scaling is motivated by the observation that by the Delta method (Dorfman, 1938), the shifted logarithm approximates the optimal variance-stabilizing transform for data from a negative binomial distribution with overdispersion $\theta = 4a$[5] (Ahlmann-Eltze and Huber, 2023). Another way to justify why log-transformed expression values make sense is by noting that the log scale emphasizes relative rather than absolute changes: On the log-scale, the ≈ 10-fold difference between 10 and 110 mRNA copies is larger ($\log_{10}(110) - \log_{10}(10) \approx 1.05$) than the same absolute difference between 1000 and 1100 copies ($\log_{10}(1100) - \log_{10}(1000) \approx 0.05$) (Amezquita et al., 2024). Therefore, after log-transform, any downstream analysis will focus on relative differences.

In practice, Equations 6-7 are often applied with adjustments specific to the use-case. E.g., L is typically chosen such that the normalized counts are on the same order of magnitude as the raw data, i.e., $L = 10\,000$ (*counts per 10k*) for UMI data, $L = 1\,000\,000$ (*counts per million*) for read count data, or simply as the median total count per cell $L = \text{median}(s_c)$ (*counts per median*) (Luecken and Theis, 2019).

When the size factors s_c are set to the total count per cell, Equation 6 is known as *library size normalization*. However, this assumes that the total count per cell is a good proxy for the true sequencing depth, and this is only true if most genes in a dataset are not differentially expressed. As this assumption might be violated in heterogeneous datasets (Lun et al., 2016), more advanced methods have been suggested to obtain size factors s_c: Lun et al. (2016) pooled similar cells to obtain more accurate size factors per cell pool, and then deconvolved these size

[5] The common pseudocount setting $a = 1$ implies quite a strong overdispersion, which often is a misspecification.

factors obtained from many pools into per-cell size factors that are also valid in the presence of many differentially expressed genes. Other methods normalize based on *spike-ins* (Jiang et al., 2011; Brennecke et al., 2013; Ding et al., 2015; Vallejos et al., 2015): equal amounts of control RNA that is added to each cell. These methods then set the size factors such that the counts for spike-in RNAs are the same in all cells after normalization — thereby equalizing between-cell differences in sequencing depth. However, one cannot easily add spike-ins in all scRNA-seq protocols, limiting the usability of this strategy (Ofengeim et al., 2017).

Some global scaling methods additionally consider per-gene effects: For protocols that sequence reads from the whole length of the RNA transcripts, Li et al. (2010) introduced *TPM normalization*[6] in which size factors take the transcript length into account (as genes with longer transcripts will result in more reads in full-length protocols). Bacher et al. (2017) observed that not all genes showed the same linear relation to the total counts per cell, i.e., some genes' counts depend more on the sequencing depth of the cell. To account for this difference in count-depth relation, the authors consequently suggested *SCnorm*, which identifies K groups of genes with similar count-depth relationship. Within each such group of similar genes, counts are then separately normalized by a size factor s_{ck} per cell c and gene group k.

Global scaling methods are very common in scRNA-seq data analysis (Luecken and Theis, 2019), likely because their basic form is simple and fast to compute (Equations 6–7). However, normalization by global scaling is not always effective: If too many genes are differentially expressed across cells, all global scaling methods that do not account for that will fail (Lun et al., 2016; Vallejos et al., 2017). Also, when cells vary a lot in their total counts, normalized and log-transformed counts can still be correlated with sequencing depth (Ahlmann-Eltze and Huber, 2023), which can introduce spurious differences between cells (Lun, 2018). Additionally, the variance stabilization can fail: The

[6] TPM (transcript per million) normalization was originally conceived for bulk RNA-seq protocols.

log-transform cannot stabilize the variance of small counts (Warton, 2018), and the common settings of a pseudocount $a = 1$ and the scaling factor $L = 1\,000\,000$ imply unrealistic amounts of overdispersion (Ahlmann-Eltze and Huber, 2023). In summary, global scaling methods are useful and popular heuristics, but their shortcomings result in measurably suboptimal performance (Cole et al., 2019; Tian et al., 2019).

Bayesian models for expression inference. Global scaling methods aim to transform raw count data such that its statistics change towards some desired properties. In particular, preprocessed counts should have stable variance across genes and be independent of the total counts per cell. Bayesian models of gene expression take a different perspective: They consider the observed matrix of mRNA counts X a noisy measurement of the inherently noisy process of gene expression, which is governed by some latent variables of biological interest, like true transcription rates T. Inferring these true transcription rates from the noisy counts is then achieved using Bayes' theorem

$$P(T|X) = \frac{P(X|T) \cdot P(T)}{P(X)}, \tag{8}$$

where $P(X|T)$ is the likelihood of the observed counts under a mechanistic model that starts from the transcriptions rates, and $P(T)$ is the prior distribution of the transcription rates. Thus, Bayesian models can infer the most likely transcription activity given some observed counts. Because the model already accounts for sources of technical and biological noise during the processes that eventually give rise to an observed mRNA count, the inferred transcription activity can be considered normalized for all noise sources that were part of the model. Also, depending on the inference procedure, the Bayesian setup usually allows to obtain uncertainty estimates for the inferred transcription activities.

Existing gene expression models differ in how much detail of the data generation they model explicitly, how they set their priors, and how they perform inference. *Normalizr* (Wang, 2021) simply assumes that ob-

served counts k_{cg} for gene g in one cell c are sampled from a Binomial distribution $k_{cg} \sim \text{Binom}(n_c, p_{gc})$ over n_c RNA molecules in the cell. *Normalizr* then estimates the $\log(p_{gc})$ as log relative expression for further processing. With a slightly more complex model that is aimed at the analysis of small sets of genes, *baredSC* (Lopez-Delisle and Delisle, 2022) models the observed counts be a Poisson sample $k_{cg} \sim \text{Poisson}(n_c \alpha_{cg})$, where n_c are the total counts in each cell. α_{cg} is the true expression and follows a Gaussian mixture model prior, where each component corresponds to a subpopulation of cells with different expression. An extension to 2D mixtures can account for correlations between two genes. However, *baredSC* obtains posterior estimates for the gene expressions α_{cg} and the number and parameters of the mixture components with Markov chain Monte Carlo (MCMC) sampling, which makes *baredSC* much slower than other methods and infeasible to use for more than a few genes of interest.

In contrast to these rather simple models, *Sanity* (Breda et al., 2021) starts from a mechanistic motivation: The methods summarizes all biological factors that influence mRNA production or decay into one gene's transcription activity a_{cg}. It then uses the relative transcription activities over all genes $\alpha_{cg} = a_{cg} / \sum_g a_{cg}$ to describe the transcriptional state of a cell that one would like to infer. *Sanity* further assumes that the number of true mRNAs in a cell are samples from a Poisson distribution that depends on α_{cg}, and that the observed scRNA-seq counts are another Poisson sample from the pool of true mRNAs. Effectively, this chain of Poisson distributions leads to a product-of-Poissons likelihood, with genes treated independently. The resulting likelihood for counts from a single gene is identical to the *baredSC* model, but as *Sanity* assumes a simple Gaussian prior over the transcription activities $alpha_{cg}$ instead of a Gaussian mixture model it cannot infer gene-gene-correlations and might oversmooth more complex multimodal distributions (Lopez-Delisle and Delisle, 2022). At the same time, *Sanity* can solve for α_{cg} faster than *baredSC*, but is still up to 4 500 times slower

than global scaling.

In summary, Bayesian models of gene expression have the advantage that they model the biophysical quantity of interest in scRNA-seq data analysis: the transcription activity. This makes the inferred normalized expression values directly interpretable. However, Bayesian inference procedures are computationally expensive, limiting the applicability of some of the models.

Null model Pearson residuals. A null model of scRNA-seq experiments describes how mRNA counts are distributed when only technical factors contribute to the data, and no biological factors are active. To use such a model for normalization and variance stabilization, one can use its Pearson residuals as normalized values:

$$R_{cg} = \frac{X_{cg} - \hat{X}_{cg,\text{null}}}{\sqrt{v_{cg,\text{null}}(X)}}, \tag{9}$$

where X_{cg} is the observed count for gene g and cell c, $\hat{X}_{cg,\text{null}}$ is the fitted null model count (i.e., the count expected under the null model given all observed counts X), and $v_{\text{cg,null}}(X)$ is the count noise expected under the null model given all observed counts X. This is equivalent to z-scoring the observed counts w.r.t. the null model. Consequently, the residuals will have mean zero and unit variance when the observed counts follow the null model perfectly. However, when the observed counts for a certain gene and cell deviate from the null model, the residuals for that count will have high absolute values. For example, when the null model assumes perfectly homogeneous counts across cells, data from cells with differential expression patterns will violate this assumption and lead to large residuals. In contrast, if the null model correctly accounts for a technical factor like sequencing depth, spurious differences in the observed counts due to sequencing depth will not lead to high residuals, as the null model accounts for them. Consequently, Pearson residuals of a well-chosen null model will (i) no longer contain expected techni-

cal variance, (ii) emphasize unexpected biological variance, and (iii) by construction stabilize the variance of all genes that are not differentially expressed. Both novel normalization methods that this thesis presents are based on null model Pearson residuals (Chapter 3–4).

Importantly, preprocessing with Pearson residuals is conceptually different from preprocessing via Bayesian models or via global scaling heuristics: Global scaling heuristics often have poor theoretical motivation, while Pearson residuals normalization states explicitly which kind of noise and expression patterns it expects and removes from the data. In contrast, when using Bayesian inference, one usually directly infers biophysically meaningful quantities that can be mapped to the transcription machinery of the cell, e.g., high transcription rates that correspond to very actively transcribed genes. In contrast, a high Pearson residual is merely a high null model z-score, indicating that some RNA count pattern significantly outside the null model's assumptions has occurred and requires interpretation by the researcher. In this sense, null model Pearson residuals can be considered an attractive, light-weight compromise between global scaling heuristics and full Bayesian inference: All assumptions relevant for data preprocessing are made explicit in the null model, while avoiding the need to model the whole biological system of interest and infer its parameters.

To set up a null model, one has to formalize its two components: the expected counts $\mathbb{E}_{\text{null}}$ and their variance Var_{null} given some observed data X. For the expected counts, one could assume that cells are identical except for their (observed) sequencing depth, and that genes do not differ except for their (observed) mean expression. Previous work by Hafemeister and Satija (2019) has in addition assumed that genes with similar mean expression should have similar dispersions. To formalize the variance of the null model, we need to describe the distribution of intrinsic and technical noise in the data.

Which noise distribution is appropriate for a null model of scRNA-seq data? The answer depends on the technical details of the sequencing

protocol that is employed to produce the data: Read counts from non-UMI protocols are sampled from the pool of cDNA *after* amplification with e.g., PCR, and contain amplification noise. In contrast, UMI counts allow to de-duplicate amplification copies, and therefore are free from such amplification noise. These protocol-level effects have been studied empirically: Grün et al. (2014) measured the magnitude of amplification noise on control data, and found that technical noise in UMI counts was on average 50% lower than in the non-UMI counts from the same experiment. Thus, different noise distributions have been established for UMI- and non-UMI data.

UMI counts are samples from the captured and reverse-transcribed RNA molecules before amplification. We can recall from above that in theory, they should follow a Poisson or negative binomial distribution. To confirm this assumption, Grün et al. (2014) studied UMI count noise empirically by investigating the variance of UMI counts across artificial replicate "cells" — samples with identical RNA content — that should contain only technical noise. They found that while UMI counts for low expression genes follow Poisson statistics, higher expression genes showed an additional noise component that required a negative binomial distribution to explain. Similarly, more recent work found biologically homogeneous UMI counts could usually be fit with a negative binomial or Poisson model (Chen et al., 2018; Svensson, 2020; Cao et al., 2021). Lopez-Delisle and Delisle (2022) suggested that these results in favor of negative binomial models did not properly take into account the cell-to-cell variance in sequencing depth, and that simple Poisson models could be sufficient.

For non-UMI data, the case is more complex. For a low-expression gene, the density of these models will often include zero, i.e., they predict that the RNA of such a gene is not observed in some cells (Figure 3a). Importantly, the amplification step differentially affects zero and non-zero counts: All non-zero counts get amplified by some (potentially noisy) multiplicative factor, trivially leading to counts that are

several magnitudes larger than UMI counts and shifting the average non-UMI count by the average amplification factor. In contrast, zero counts are not amplified and therefore do not shift (Figure 3c).

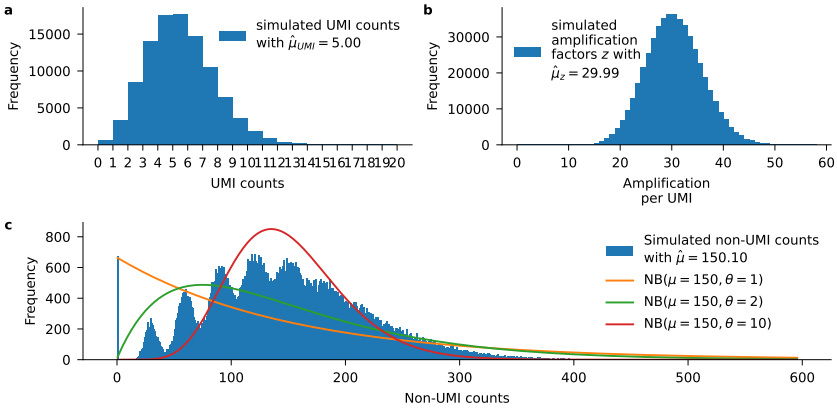

Figure 3: Amplification of Poisson-distributed UMIs leads to zero-inflated non-UMI counts.
a: Simulated UMI counts from a Poisson with $\mu = 5$, mimicking the pre-amplification counts for a low-expression gene with some true zeros. **b:** Simulated noisy amplification factors z from a Poisson with $\mu_z = 30$ **c:** Simulated non-UMI counts: For each UMI count k in (a), we draw k amplification factors z from (b) and sum them up to arrive at non-UMI counts $x = \sum_{i=1}^{k} z_i$. See Section 4.2.2 for model details. Note the spike at zero corresponding to zero UMIs, and the modes at multiples of μ_z corresponding to the nonzero UMI counts 1, 2, 3, ... after amplification. Lines show probability mass functions for different negative binomial distributions with the correct mean $\mu_z \cdot \mu = 150$ and different overdispersion, none of which can fit zeros and nonzeros at the same time. Note that a Poisson distribution (not shown) would fit even worse, as it is even narrower around the mean than any negative binomial.

The resulting non-UMI counts therefore exhibit zeros *and* a mixture of shifted Poisson or negative binomial distributions (Figure 3c), leading to a characteristic gap in their density between zeros and shifted non-zeros. Such a gap is not present in Poisson or negative binomial densities, and the zeros would be unexpected for any Poisson or negative binomial with the new, shifted mean. As a result, just changing the mean of these distributions to adjust for the shift will be insufficient because

of the excess zeros(Figure 3c). That is why non-UMI counts are often described as *zero-inflated*. Previous work has attempted to model them as such, e.g., by combining distribution for the non-zero counts with an additional probability mass at or around zero (Kharchenko et al., 2014; Finak et al., 2015; Lopez et al., 2018; Wu et al., 2021). A typical choice for non-UMI noise is the zero-inflated negative binomial (ZINB), which is a negative binomial with an additional parameter ψ for the probability that a count is not originating from the NB but from a separate point mass at zero, leading to the PMF:

$$P_{\mathrm{ZINB}}(X) = \begin{cases} \psi + (1-\psi) \cdot P_{\mathrm{NB}}(0, \mu, \theta) & \text{for } x = 0 \\ (1-\psi) \cdot P_{\mathrm{NB}}(x, \mu, \theta) & \text{for } x > 0 \end{cases}. \tag{10}$$

Empirically, Chen et al. (2018) showed that in non-UMI data, a substantial fraction of genes required a ZINB model to be fit well, and Cao et al. (2021) confirmed that non-UMI data are zero-inflated regardless of the technical platform they are obtained from.

In summary, null models of UMI data should use Poisson or negative binomial noise, while null models for non-UMI data require additional zero-inflation as in the ZINB model. For UMI data, several residuals-based approaches exist that turn this knowledge into null models: Hafemeister and Satija (2019) use a negative binomial regression null model, which allows groups of genes with a similar mean expression to behave similar (i.e., have a similar relation to each cell's sequencing depth and a similar overdispersion). Townes et al. (2019) suggest deviance residuals in the framework of generalized PCA, using similar distributional assumptions at its core. Very recently, Singh and Khiabanian (2024) suggested a residual-based approach similar to Hafemeister and Satija (2019), that improved sequencing depth estimation by relying only on genes that are not differentially expressed. In contrast, for non-UMI data, a comparable null model residual approach does not yet exist.

2.3.2 Visualization

Feature selection. Raw single-cell RNA-seq data is high-dimensional, as typical datasets contain the expression of ten thousands of genes. However, usually only a small subset of genes is biologically informative, while the rest is noisy or not differentially expressed (Amezquita et al., 2020). To remove these genes and reduce the computational cost, many preprocessing pipelines contain a feature selection step to select only highly variable genes for further analysis (Yip et al., 2019; Luecken and Theis, 2019; Heumos et al., 2023). This is often done by selecting *highly variable genes (HVGs)* that have high variance after preprocessing, assuming that their variance is mostly due to biologically interesting causes. This step requires proper variance stabilization because otherwise only high-expression genes will dominate (Amezquita et al. (2020), see Section 2.3.1). To avoid this, some HVG selection methods directly use null model residuals as described above (Townes et al., 2019; Hafemeister and Satija, 2019).

Linear dimensionality reduction: PCA. It is best practice to rather select too many than too few HVGs, often on the order of hundreds to thousands of genes (Luecken and Theis, 2019), so by the number of features, the dimensionality of the data is still high after feature selection. Fortunately, Heimberg et al. (2016) showed in an analysis of hundreds of scRNA-seq datasets that their intrinsic dimensionality is much lower, as many genes are correlated. This motivates the use of principal component analysis (PCA) for linear dimensionality reduction. Informally, one can describe PCA rotating the high-dimensional gene expression space such that the directions with maximum variance align with the first cardinal axes. Effectively, this can summarize many correlated dimensions into one principal axis. To reduce dimensionality, one can now select a small number of these principal axes that explain most of the variance. This compresses the data into a low-rank representation that implicitly removes noise by pooling redundant genes (Luecken

and Theis, 2019; Amezquita et al., 2020). As PCA is a linear method, the distances in PCA space remain interpretable (Luecken and Theis, 2019), and one can map principal components and their combinations back to gene space (Chung and Storey, 2014), e.g., for biological interpretations of a principal component as set of genes that act jointly in gene regulatory networks. While classic PCA implicitly assumes that the data in the original space follows a multivariate normal distribution, Townes et al. (2019) proposed to adjust PCA to scRNA-seq data and introduced GLM-PCA, a variation of PCA that assumes Poisson or negative binomial count data instead.

Non-linear dimensionality reduction: *t*-SNE and UMAP. Reducing dimensionality with PCA after preprocessing scRNA-seq data leads to a useful space for many downstream analyses like clustering (Moon et al., 2018). However, usually the variance of the data spreads over more than 2 principal axes (often between 10 and 50), making it hard to visualize even the reduced PCA space directly. To address this problem, non-linear dimensionality reduction methods like *t*-SNE (van der Maaten and Hinton, 2008) and UMAP (McInnes et al., 2018) became very popular to visualize scRNA-seq datasets in two dimensions (Kobak and Berens, 2019; Becht et al., 2019). These methods are governed by an objective function that encourages similar cells from the high-dimensional space (e.g., the PCA space after scRNA-seq preprocessing) to stay close in the low dimensional 2D embedding. Practically, *t*-SNE starts by identifying nearest neighbors among the cells in the high-dimensional space. Then, all cells are assigned an initial position in 2D space[7]. These initial cell positions are then iteratively updated with gradient descent on the objective function, which moves cells such that similar cells come closer together. To see how this is achieved, one can decompose the gradient

[7] Usually, the first two principal components are used for initialization (Poličar et al., 2024), as they are a good proxy for the global structure in the data. As *t*-SNE is an inherently local method that only optimizes nearest-neighbor relations, it cannot produce such global structure by itself and therefore depends on a meaningful initialization (Kobak and Berens, 2019).

forces acting on each cell into attractive and repulsive forces (van der Maaten and Hinton, 2008): The attractive forces are only between the original nearest neighbors from the high-dimensional space, i.e., similar cells attract each other. In contrast, repulsive forces act between all cells. When these forces are applied many times, similar cells will usually group into well-separated clusters.

The balance of these attractive and repulsive forces is the defining feature of UMAP, t-SNE, and related methods. Recent insights by Böhm et al. (2022) revealed that this perspective unifies many non-linear dimensionality reduction methods and explains the differences in the embeddings they produce. This can be helpful for scRNA-seq analysis as well, as different settings highlight different aspects of the data (Damrich et al., 2024): High attraction methods tend to emphasize continuous structure in the data (like developmental trajectories), while high repulsion brings out discrete clusters and local structure.

t-SNE's ability to resolve high-dimensional clusters is one of its strengths in practice (Kobak and Berens, 2019) and was studied in a line of theoretical analysis: Linderman and Steinerberger (2017) found hyperparameter settings for which well-separated clusters will always be correctly recovered. Arora et al. (2018) improved on this result by showing which kinds of clusters can be fully or partially recovered. More recently, Cai and Ma (2022) have offered a whole theoretical framework to explain which parts of the t-SNE algorithm give rise to its good performance in practice. In particular, they find that t-SNE is inherently related to spectral clustering (Von Luxburg, 2007), explaining its ability to flexibly cluster data with an unknown number of classes.

In contrast to these advantages, t-SNE and UMAP embeddings also have well-known shortcomings (Wattenberg et al., 2016; Nonato and Aupetit, 2018; Wang et al., 2023b; Chari and Pachter, 2023): The objective function focuses solely on placing neighboring cells from the high-dimensional space close to each other in the 2D embedding, and does not attempt to respect the original distances between points. Thus, the

cell-to-cell distances in the final embedding will usually not resemble the original, high-dimensional distances. This partly follows from the objectives of t-SNE and UMAP that simply not include distance preservation, but also goes back to fundamental problems in the mathematics of dimensionality reduction: Reducing dimensionality inevitably leads to loss of information. Specifically, some point configurations that are possible in high dimensions cannot be accommodated in lower dimensions. For example, consider a k-simplex: It consists of $k+1$ points that are equidistant to each other in a k-dimensional space, and cannot be embedded in fewer than k dimensions.[8] In practice, five cells that are approximately equidistant to each other in the high-dimensional gene expression space form a 4-simplex, and require at least four dimensions to be embedded without distorting the equidistance. Consequently, a 2D t-SNE or UMAP would necessarily distort the distances of such equidistant cells. Recently, Chari and Pachter (2023) have shown that such cases of severe distance distortion exist in real scRNA-seq datasets. As a result, distances on a t-SNE or UMAP embedding can usually not be interpreted.

In summary, t-SNE and UMAP are unable to preserve distances, but have proven useful to recover local information and cluster structure. For practitioners, the advantages of having a single-plot summary visualization of their data seems to outweigh the shortcoming of not preserving all aspects of the dataset perfectly: t-SNE and UMAP have grown to be a standard tool in visualizing scRNA-seq data (Luecken and Theis, 2019; Amezquita et al., 2020).

[8] A closely related result is the curse of dimensionality for distances, stating that generally, distances between n points grow with the number of dimensions p and become more and more similar (Aggarwal et al., 2001; Altman and Krzywinski, 2018), with all points being far away from each other for many dimensions. If one applies dimensionality reduction to such a space, one cannot ensure anymore that all n points are far away—especially when embedding large n into very few dimensions. As a result, one will inevitably need to embed some points closer to each other than they were in the original dimensions.

2.3.3 Open questions and outlook

Preprocessing scRNA-seq data. Section 2.3.1 demonstrated the diversity in preprocessing methods for scRNA-seq data. Users can choose from a spectrum that ranges from fast and simple heuristics (global scaling normalization and log-transform) to rather slow Bayesian models of the scRNA-seq data generation process. Recently, Pearson residuals of null models emerged as a promising alternative that can be seen as a compromise between the two.

However, the setup of the most popular residual-based preprocessing approach, *scTransform* (Hafemeister and Satija, 2019), raises critical questions: Their null model (Equation 16) is obtained by fitting a negative binomial regression with a large number of parameters (three parameters per gene). Even though these fits are regularized (i.e., fitted parameters are shared between groups of genes with similar mean expression), this leads to an expensive setting with many free parameters, and it is unclear if this complexity is really needed. Also, the fitted gene parameters appear to be correlated with the mean expression and among each other (Figure 4) — a behavior that is not motivated by theory and requires investigation. We will explore these open questions about the *scTransform* model in Chapter 3 (Lause et al., 2021), and propose a simpler and faster residual-based preprocessing for UMI data: Analytic Pearson residuals.

All recent work on residual-based preprocessing is based on null models for UMI data (Hafemeister and Satija, 2019; Townes et al., 2019; Singh and Khiabanian, 2024); null models for non-UMI data are currently not available. As non-UMI data from full-length protocols are still regularly used to study splicing and gene isoforms, we suggest to close this gap: Chapter 4 (Lause et al., 2023) develops an appropriate null model for non-UMI counts and extends the Analytic Pearson residuals from Chapter 3 to non-UMI data.

Visualizing scRNA-seq data. After preprocessing, researchers usually want to inspect their data visually. Section 2.3.2 motivated the de-facto standard pipeline for visualization of high-dimensional single-cell data : HVG selection and PCA followed by UMAP or t-SNE. While HVG selection and PCA are undisputed tools that remove uninformative genes and pooling correlated dimensions, the use of UMAP and t-SNE has recently been challenged: Chari and Pachter (2023) re-analyzed how well these 2D embedding methods preserve local, global and distance information from the high-dimensional space. The authors conclude that UMAP and t-SNE fail at each one of these tasks and even claim that the resulting 2D embeddings are "arbitrary" and should not be used at all.

This conclusion is surprising and in stark contrast to the popularity of 2D embeddings in the field of single-cell biology: While some shortcomings regarding distance preservation are well known (see above), Chari and Pachter (2023) themselves note that 2D embeddings are commonly used for sanity-checking preprocessing and exploratory data analysis. Additionally, a line of theoretical results indicates that t-SNE can even guarantee to recover well-separated ground truth cluster structure (Arora et al., 2018; Linderman and Steinerberger, 2019; Cai and Ma, 2022). How can this be true if t-SNE and UMAP are "arbitrary"?

The final Chapter 5 of this thesis resolves this contradiction: We show that a core part of the Chari and Pachter (2023) argument for "arbitrary" embeddings is flawed, as they evaluate embedding quality only through distance preservation. Our analysis demonstrates that t-SNE and UMAP perform well on cluster- and neighborhood preservation, and we conclude that all things considered, 2D embeddings remain a useful tool for exploratory single-cell data analysis.

3 Analytic Pearson residuals for UMI counts

Publication note. This chapter is published in *Genome Biology* (Lause et al., 2021) and is available at `https://doi.org/10.1186/s13059-021-02451-7` under a `CC-BY 4.0` license (`https://creativecommons.org/licenses/by/4.0/`).

Abstract

Standard preprocessing of single-cell RNA-seq UMI data includes normalization by sequencing depth to remove this technical variability, and nonlinear transformation to stabilize the variance across genes with different expression levels. Instead, two recent papers propose to use statistical count models for these tasks: Hafemeister & Satija (Hafemeister and Satija, 2019) recommend using Pearson residuals from negative binomial regression, while Townes et al. (Townes et al., 2019) recommend fitting a generalized PCA model. Here, we investigate the connection between these approaches theoretically and empirically, and compare their effects on downstream processing.

We show that the model of Hafemeister and Satija produces noisy parameter estimates because it is overspecified, which is why the original paper employs post-hoc smoothing. When specified more parsimoniously, it has a simple analytic solution equivalent to the rank-one Poisson GLM-PCA of Townes et al. Further, our analysis indicates that per-gene overdispersion estimates in Hafemeister and Satija are biased, and that the data are in fact consistent with the overdispersion parameter being independent of gene expression. We then use negative control data without biological variability to estimate the technical overdispersion of UMI counts, and find that across several different experimental protocols, the data are close to Poisson and suggest very moderate overdispersion. Finally, we perform a benchmark to compare the performance of Pearson residuals, variance-stabilizing transformations, and GLM-PCA on scRNA-seq datasets with known ground truth.

> We demonstrate that analytic Pearson residuals strongly outperform other methods for identifying biologically variable genes, and capture more of the biologically meaningful variation when used for dimensionality reduction.

3.1 Introduction

The standard preprocessing pipeline for single-cell RNA-seq data includes sequencing depth normalization followed by log-transformation (Luecken and Theis, 2019; Amezquita et al., 2020). The normalization aims to remove technical variability associated with cell-to-cell differences in sequencing depth, whereas the log-transformation is supposed to make the variance of gene counts approximately independent of the mean expression. Two recent papers argue that neither step works very well in practice (Hafemeister and Satija, 2019; Townes et al., 2019). Instead, both papers suggest to model UMI (unique molecular identifier) data with count models, explicitly accounting for the cell-to-cell variation in sequencing depth (defined here as the total UMI count per cell). Hafemeister and Satija (2019) use a negative binomial (NB) regression model (`scTransform` package in R), while Townes et al. (2019) propose Poisson generalized principal component analysis (GLM-PCA). These two models are seemingly very different.

Here we show that the model used by Hafemeister and Satija (2019) has a too flexible parametrization, resulting in noisy parameter estimates. As a consequence, the original paper employs post-hoc smoothing to correct for that. We show that a more parsimonious model produces stable estimates even without smoothing and is equivalent to a special case of GLM-PCA. We then demonstrate that the estimates of gene-specific overdispersion in the original paper are strongly biased, and further argue that UMI data do not require gene-specific overdispersion parameters to account for technical noise. Rather, the technical variability is consistent with the same overdispersion parameter shared between all genes. We use available negative control datasets to estimate this

technical overdispersion. Furthermore, we compare Pearson residuals, GLM-PCA, and variance-stabilizing transformations for highly variable gene selection and as data transformation for downstream processing.

Our code in Python is available at `http://github.com/berenslab/umi-normalization`. Analytic Pearson residuals will be included into upcoming `Scanpy` 1.9 (Wolf et al., 2018).

3.2 Results

3.2.1 Analytic Pearson residuals

A common modeling assumption for UMI or read count data without biological variability is that each gene g takes up a certain fraction p_g of the total amount n_c of counts in cell c (Love et al., 2014; Eling et al., 2018; Lopez et al., 2018; Townes et al., 2019; Svensson et al., 2020b; Sarkar and Stephens, 2021). The observed UMI counts X_{cg} are then modelled as Poisson or negative binomial (NB) (Grün et al., 2014) samples with expected value $\mu_{cg} = p_g n_c$ without zero-inflation (Svensson, 2020; Sarkar and Stephens, 2021):

$$X_{cg} \sim \text{Poisson}(\mu_{cg}) \text{ or } \text{NB}(\mu_{cg}, \theta), \tag{11}$$

$$\mu_{cg} = n_c p_g. \tag{12}$$

The Poisson model has a maximum likelihood solution (see Methods) that can be written in closed form as $\hat{n}_c = \sum_g X_{cg}$ (sequencing depths), $\hat{p}_g = \sum_c X_{cg} / \sum_c \hat{n}_c$, or, put together,

$$\hat{\mu}_{cg} = \frac{\sum_j X_{cj} \cdot \sum_i X_{ig}}{\sum_{ij} X_{ij}} \tag{13}$$

For the negative binomial model this holds only approximately. Using this solution, the Pearson residuals are given by

$$Z_{cg} = \frac{X_{cg} - \hat{\mu}_{cg}}{\sqrt{\hat{\mu}_{cg} + \hat{\mu}_{cg}^2/\theta}}, \tag{14}$$

where $\mu_{cg} + \mu_{cg}^2/\theta$ is the NB variance and $\theta \to \infty$ gives the Poisson limit. The variance of Pearson residuals is, up to a constant, equal to the Pearson χ^2 goodness-of-fit statistic (Agresti, 2015) and quantifies how much each gene deviates from this constant-expression model. As pointed out by Aedin Culhane (Culhane, 2020), singular value decomposition of the Pearson residuals under the Poisson model is known as *correspondence analysis* (Hill, 1974; Greenacre and Hastie, 1987; Greenacre, 2007; Holmes, 2008), a method with a longstanding history (Hirschfeld, 1935).

Hafemeister and Satija (2019) suggested using Pearson residuals from a related NB regression model for highly variable gene (HVG) selection and also as a data transformation for downstream processing. In parallel, Townes et al. (2019) suggested using deviance residuals (see Methods) from the same Poisson model as above for HVG selection and also for PCA as an approximation to their GLM-PCA. In the next sections we discuss the relationships between these approaches.

3.2.2 The regression model in `scTransform` is overspecified

Hafemeister and Satija (2019) used the 33k PBMC (peripheral blood mononuclear cells, an immune cell class that features several distinct subpopulations) dataset from 10X Genomics in their work on normalization of UMI datasets. For each gene g in this dataset, the authors fit an independent NB regression

$$X_{cg} \sim \mathrm{NB}(\mu_{cg}, \theta_g) \tag{15}$$

$$\ln(\mu_{cg}) = \beta_{0g} + \beta_{1g} \log_{10}(\hat{n}_c). \tag{16}$$

Here θ_g is the gene-specific overdispersion parameter, $\hat{n}_c$ are observed sequencing depths as defined above, and β_{0g} and β_{1g} are the gene-specific intercept and slope. The natural logarithm follows from the logarithmic link function that is used in NB regression by default. The original paper estimates β_{0g} and β_{1g} using Poisson regression, and then uses the obtained estimates to find the maximum likelihood estimate of θ_g. The

resulting estimates for each gene are shown in Figure 4a–c, reproducing Figure 2A from the original paper.

The authors observed that the estimates $\hat{\beta}_{0g}$ and $\hat{\beta}_{1g}$ were unstable

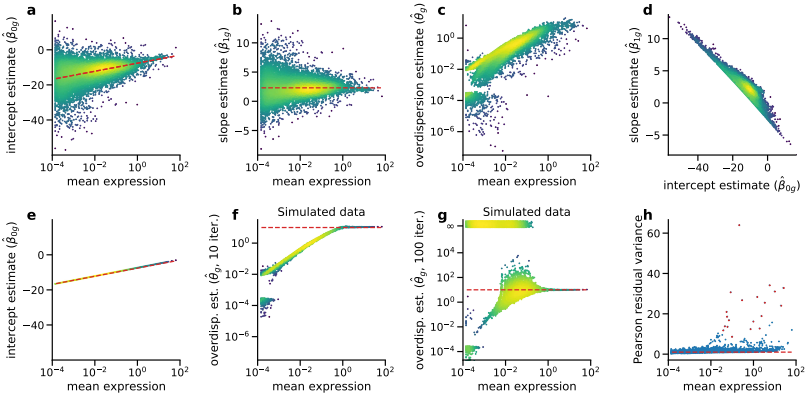

Figure 4: Regression model of Hafemeister and Satija (2019) compared to the offset model. Each dot corresponds to a model fit to the counts of a single gene in the 33k PBMC dataset (10x Genomics, $n = 33\,148$ cells). Following Hafemeister and Satija (2019), we included only the 16 809 genes that were detected in at least five cells. Color denotes the local point density from low (blue) to high (yellow). Expression mean was computed as $\frac{1}{n}\sum_c X_{cg}$. **a:** Intercept estimates $\hat{\beta}_{0g}$ in the original regression model. Dashed line: Analytic solution for $\hat{\beta}_{0g}$ in the offset model we propose. **b:** Slope estimates $\hat{\beta}_{1g}$. Dashed line: $\beta_{1g} = \ln(10) \approx 2.3$. **c:** Overdispersion estimates $\hat{\theta}_g$. **d:** Relationship between slope and intercept estimates ($\rho = -0.91$). **e:** Intercept estimates in the offset model, where the slope coefficient is fixed to 1. Dashed line shows the analytic solution, which is a linear function of gene mean. **f:** Overdispersion estimates $\hat{\theta}_g$ on simulated data with true $\theta = 10$ (dashed line) for all genes. **g:** Overdispersion estimates $\hat{\theta}_g$ on the same simulated data as in panel (f), but now with 100 instead of 10 iterations in the `theta.ml()` optimizer (R, `MASS` package). Cases for which the optimization diverged to infinity or resulted in spuriously large estimates ($\hat{\theta}_g > 10^6$) are shown at $\hat{\theta}_g = \infty$ with some jitter. Dashed line: true value $\theta = 10$. **h:** Variance of Pearson residuals in the offset model. The residuals were computed analytically, assuming $\theta = 100$ for all genes. Following Hafemeister and Satija (2019), we clipped the residuals to a maximum value of $\sqrt{n}$. Dashed line indicates unit variance. Red dots show the genes identified in the original paper as most variable.

and showed high variance for genes with low average expression (Figure 4a–b). They addressed this with a 'regularization' procedure that re-set all estimates to the local kernel average estimate for a given expression level. This is similar to some approaches to bulk RNA-seq analysis (Love et al., 2014; Eling et al., 2018) but with post-hoc correction instead of Bayesian shrinkage. This kernel smoothing resulted in an approximately linear increase of the intercept with the logarithm of the average gene expression (Figure 4a) and an approximately constant slope value of $\hat{\beta}_{1g} \approx 2.3$ (Figure 4b). The nature of these dependencies was left unexplained. Moreover, we found that $\hat{\beta}_{0g}$ and $\hat{\beta}_{1g}$ were strongly correlated ($\rho = -0.91$), especially for weakly expressed genes (Figure 4d). Together, these clear symptoms of overfitting suggest that the regression model was overspecified.

Indeed, the theory calls for a less flexible model.As explained above, a common modeling assumption (Eq. 12) is that $\mu_{cg} = p_g n_c$, or equivalently

$$\ln(\mu_{cg}) = \ln(p_g) + \ln(n_c) = \beta_{0g} + \ln(n_c). \tag{17}$$

We see that under this assumption, the slope β_{1g} does not need to be fit at all and should be fixed to 1, if $\ln(n_c)$ is used as predictor. Not only does this suggest an alternative, simpler parametrization of the model, but it also explains why Hafemeister and Satija (2019) found that $\hat{\beta}_{1g} \approx 2.3$: they used $\log_{10}(n_c) = \ln(n_c)/\ln(10)$ instead of $\ln(n_c)$ as predictor, and so obtained $\ln(10) \approx 2.3$ as the average slope.

Under the assumption of Eq. 17, a Poisson or NB regression model should be specified using $\ln(n_c)$ as predictor with a fixed slope of 1, a so-called *offset* (Eqs. 15 and 17). This way, the resulting model has only one free parameter and is not overspecified. Moreover, the Poisson offset model is equivalent to Eqs. 11–12 and so, as explained above, has an analytic solution

$$\hat{\beta}_{0g} = \ln\left(\sum_c X_{cg} / \sum_c n_c\right) = \ln(\tfrac{1}{N}\sum_c X_{cg}) - \ln(\tfrac{1}{N}\sum_c n_c), \tag{18}$$

which forms a straight line when plotted against the log-transformed average gene expression $\frac{1}{n}\sum_c X_{cg}$ (Figure 4e). This provides an explanation for the linear trend in $\hat{\beta}_{0g}$ in the original two-parameter model (Figure 4a).

In practice, our one-parameter offset model and the original two-parameter model after smoothing arrive at qualitatively similar results (Figure 4h). However, we argue that the one-parameter model is more appealing from a theoretical perspective, has an analytic solution, and does not require post-hoc averaging of the coefficients across genes.

3.2.3 The offset regression model is equivalent to the rank-one GLM-PCA

The offset regression model turns out to be a special case of GLM-PCA (Townes et al., 2019). There, the UMI counts are modeled as

$$X_{cg} \sim \text{Poisson}(\mu_{cg}) \text{ or } \text{NB}(\mu_{cg}, \theta_g), \tag{19}$$

$$\mu_{cg} = n_c \exp\big(\sum_{l=0}^{k} U_{cl}V_{lg}\big) = n_c \exp\big(V_{0g} + \sum_{l=1}^{k} U_{cl}V_{lg}\big), \tag{20}$$

assuming $k+1$ latent factors, with U and V playing the role of principal components and corresponding eigenvectors in standard PCA. Importantly, the first latent factor is constrained to $U_{c0} = 1$ for all cells c, such that V_{0g} can be interpreted as gene-specific intercepts. If the data are modeled without any further latent factors, Eq. 20 reduces to

$$\ln(\mu_{cg}) = V_{0g} + \ln(n_c), \tag{21}$$

which is identical to Eq. 17 with $V_{0g} = \beta_{0g}$. This shows that the proposed one-parameter offset regression model is exactly equivalent to the intercept-only rank-one GLM-PCA.

3.2.4 Overdispersion estimates in `scTransform` are biased

After discussing the overparametrization of the systematic component of the `scTransform` model, we now turn to the NB noise model employed by Hafemeister and Satija (2019). The $\hat{\theta}_g$ estimates in the original paper are monotonically increasing with the average gene expression, both before and after kernel smoothing (Figure 4c). This suggests that there is a biologically meaningful relationship between the expression strength and the overdispersion parameter θ_g. However, this conclusion is in fact unsupported by the data.

To demonstrate this, we simulated a dataset with NB-distributed counts $\widetilde{X}_{cg} \sim \mathrm{NB}(\mu_{cg}, \theta = 10)$ with μ_{cg} given by Eq. 13 using X_{cg} of the PBMC dataset. Applying the original estimation procedure to this simulated dataset showed the same positive correlation of $\hat{\theta}_g$ with the average expression as in real data (Figure 4f), strongly suggesting that it does not represent an underlying technical or biological cause, but only the estimation bias. Low-expressed genes had a larger bias and only for genes with the highest average expression was the true $\theta = 10$ estimated correctly.

Moreover, the $\hat{\theta}_g$ estimates strongly depended on the exact details of the estimation procedure. Using the `theta.ml()` R function with its default 10 iterations, as Hafemeister and Satija (2019) did, led to multiple convergence warnings for the simulated data in Figure 4f. Increasing this maximum number of iterations to 100 eliminated most convergence warnings, but caused 49.9% of the estimates to diverge to infinity or above 10^{10} (Figure 4g). These instabilities are likely due to shallow maxima in the NB likelihood w.r.t. θ (Willson et al., 1986).

The above arguments show that the overdispersion parameter estimates in Hafemeister and Satija (2019) for genes with low expression were strongly biased. In practice, however, the predicted variance $\mu + \mu^2/\theta$ is only weakly affected by the exact value of θ for low expression means μ, and so the bias reported here does not substantially affect the Pearson residuals (see below). Also, many of the weakly expressed genes

may be filtered out during preprocessing in actual applications. We note that large errors in NB overdispersion parameter estimates have been extensively described in other fields, with simulation studies showing that estimation bias occurs especially for low NB means, small sample sizes, and large true values of θ (Clark and Perry, 1989; Lord, 2006; Lord and Miranda-Moreno, 2008), i.e., for samples that are close to the Poisson distribution. Note also that post-hoc smoothing (Hafemeister and Satija, 2019) can reduce the variance of the $\hat{\theta}_g$ estimates, but does not reduce the bias.

3.2.5 Negative control datasets suggest low overdispersion

To avoid noisy and biased estimates, we suggest to use one common θ value shared between all genes. Of course, any given dataset would be better fit using gene-specific values θ_g. However, our goal is not the best possible fit: We want the model to account only for technical variability, but not biological variability, e.g., between cell types; this kind of variability should manifest itself as high residual variance.

Rather than estimating the θ value from a biologically heterogeneous dataset such as PBMC, we think it is more appropriate to estimate the technical overdispersion using negative control datasets, collected without any biological variability (Svensson, 2020). We analyzed several such datasets spanning different droplet- and plate-based sequencing protocols (10x Genomics, inDrop, MicrowellSeq) and compared the $\hat{\theta}_g$ estimates to the estimates obtained using simulated NB data with various known values of $\theta \in \{10, 100, 1000, \infty\}$. For the simulations, we used the empirically observed sample sizes and sequencing depths. We found that across different protocols, negative control data were consistent with overdispersion $\theta \approx 100$ or larger (Figure S1). The plateau at $\theta \approx 10$ in the PBMC data visible in Figure 4c could reflect biological and not technical variability. At the same time, negative control data were not consistent with the Poisson model ($\theta = \infty$), but likely overdispersion parameter values ($\theta \approx 100$) are large enough to make the Poisson model

acceptable in practice (Kim et al., 2015; Wang et al., 2018; Sarkar and Stephens, 2021). Parallel work reached the same conclusion (Lopez-Delisle and Delisle, 2022).

3.2.6 Analytic Pearson residuals select biologically relevant genes

Both Hafemeister and Satija (2019) and Townes et al. (2019) suggested to use Pearson/deviance residuals based on models that only account for technical variability, in order to identify biologically variable genes. Indeed, genes showing biological variability should have higher variance than predicted by such a model. As explained above, Pearson residuals in the model given by Eqs. 11–12 (or, equivalently, offset regression model or rank-one GLM-PCA) can be conveniently written in closed form:

$$Z_{cg} = \frac{X_{cg} - \hat{\mu}_{cg}}{\sqrt{\hat{\mu}_{cg} + \hat{\mu}_{cg}^2/\theta}}, \quad \hat{\mu}_{cg} = \frac{\sum_j X_{cj} \cdot \sum_i X_{ig}}{\sum_{i,j} X_{ij}}, \quad \theta = 100. \tag{22}$$

For most genes in the PBMC data, the variance of the Pearson residuals was close to 1, indicating that this model predicted the variance of the data correctly and suggesting that most genes did not show biological variability (Figures 4h). Using $\theta = 100$ led to several high-expression genes selected as biologically variable that would not be selected with a lower θ (e.g., *Malat1*), but overall, using $\theta = 10$, $\theta = 100$, or even the Poisson model with $\theta = \infty$ led to only minor differences (Figure S2). Using analytic Pearson residuals for HVG selection yielded a very similar result compared to using Pearson residuals from the smoothed regression presented in Hafemeister and Satija (2019), with almost the same set of genes identified as biologically variable (Figures 4h, S2). This suggests that our model is sufficient to identify biologically relevant genes.

It is instructive to compare the variance of Pearson residuals to the variance that one gets after explicit sequencing depth normalization followed by a variance-stabilizing transformation. For Poisson data, the square root transformation $\sqrt{x}$ is approximately variance-stabilizing,

and several modifications exist in the literature (Bar-Lev and Enis, 1988), such as the Anscombe transformation $2\sqrt{x + 3/8}$ (Anscombe, 1948) and the Freeman-Tukey transformation $\sqrt{x} + \sqrt{x + 1}$ (Freeman and Tukey, 1950). Normalizing UMI counts X_{cg} by sequencing depths n_c (and multiplying the result by the median sequencing depth $\langle n_c \rangle$ across all cells; 'median normalization') followed by one of the square-root transformations has been advocated for UMI data processing (Wagner, 2019, 2020).

Comparing the gene variances after the square-root transformation (Figure 5a) with those of Pearson residuals (Figure 5b) in the PBMC dataset showed that the square-root transformation is not sufficient for variance stabilization. Particularly affected are low-expression genes that have variance close to zero after the square-root transform (Warton, 2018). For example, platelet gene markers such as *Tubb1* have low average expression (because platelets are a rare population in the PBMC dataset) and do not show high variance after any kind of square-root transform (another example was given by the B-cell marker *Cd79a*). At the same time, Pearson residuals correctly indicate that these genes have high variance and are biologically meaningful (Figure 5c). For the genes with higher average expression, some differentially expressed genes like the monocyte marker *Lyz* or the above-mentioned *Malat1* showed high variance in both approaches. However, the selection based on the square-root transform also included high-expression genes like *Fos*, which showed noisy and biologically unspecific expression patterns (Figure 5c). Similar patterns were observed in the full-retinal dataset (Macosko et al., 2015) (Figure S3).

The gene with the highest average expression in the PBMC dataset, *Malat1*, showed clear signs of biologically meaningful variability: e.g., it is not expressed in platelets (Figure 5c). While this gene is selected as biologically variable based on Pearson residuals with $\theta \approx 100$ as we propose (Figure 5b), it was not selected by Hafemeister and Satija (2019) who effectively used $\theta \approx 10$ (Figures 4c,h, S2). This again suggests that

$\theta \approx 100$ is more appropriate than $\theta \approx 10$ to model technical variability of UMI counts.

Pearson residuals may even be 'too sensitive' in that genes that are only expressed in a handful cells may get very large residual variance. Hafemeister and Satija (2019) suggested clipping residuals to $[-\sqrt{n}, \sqrt{n}]$. We found that this step avoids large residual variance in very weakly expressed genes (Figure S2 see Methods for more details). The variance of unclipped Pearson residuals under the Poisson model ($\theta = \infty$) was very similar to the Fano factor of counts after median normalization (Figure S2) and less useful for HVG selection compared to the clipped residuals.

Lastly, gene selection by the widely-used $\log(1+x)$-transform as well

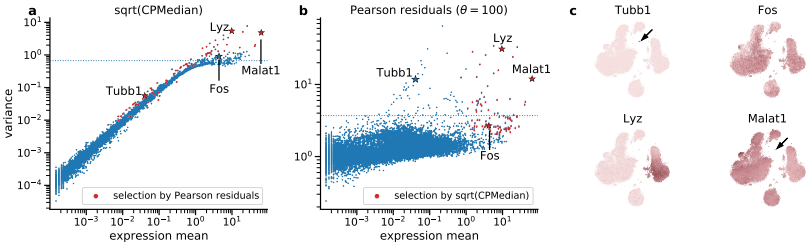

Figure 5: Selection of variable genes. In the first two panels, each dot shows the variance of a single gene in the PBMC dataset after applying a normalization method. The dotted horizontal line shows a threshold adjusted to select 100 most variable genes. Red dots mark 100 genes that are selected by the other method, i.e., that are above the threshold in the other panel. Stars indicate genes shown in the last panel. **a:** Gene variance after sequencing depth normalization, median-scaling, and the square-root transformation. **b:** Variance of Pearson residuals (assuming $\theta = 100$). **c:** t-SNE of the entire PBMC dataset (see Figure S4), colored by expression of four example genes (after sequencing depth normalization and square-root transform). Platelet marker *Tubb1* with low average expression is only selected by Pearson residuals. Arrows indicate the platelet cluster. *Fos* is only selected by the square root-based method, and does not show a clear type-specific expression pattern. *Malat1* (expressed everywhere apart from platelets) and monocyte marker *Lyz* with higher average expression are selected by both methods.

as by the variance of deviance residuals as suggested by Townes et al. (2019) led to very similar results as described above for the square-root transform: many biologically meaningful genes were not selected, as all three methods overly favored high-expression genes (Figure S2). In conclusion, neither of these transformations is sufficiently variance-stabilizing. In practice, many existing HVG selection methods take the mean-variance relationship into account when performing the selection (e.g., `seurat` and `seurat_v3` methods (Satija et al., 2015; Stuart et al., 2019) as implemented in `Scanpy` (Wolf et al., 2018)). We benchmarked their performance in the next section.

3.2.7 Analytic Pearson residuals separate cell types better than other methods

Next, we studied the effect of different normalization approaches on PCA representations and t-SNE embeddings. The first approach is median normalization, followed by the square-root transform (Wagner, 2019, 2020). We used 50 principal components of the resulting data matrix to construct a t-SNE embedding. The second approach is computing Pearson residuals according to Eq. 22 with $\theta = 100$, followed by PCA reduction to 50 components. The third approach is computing 50 components of negative binomial GLM-PCA with $\theta = 100$ (Townes et al., 2019). We used the same initialization to construct all t-SNE embeddings to ease the visual comparison (Kobak and Berens, 2019).

We applied these methods to the full PBMC dataset (Figure S4), three retinal datasets (Macosko et al., 2015; Shekhar et al., 2016; Tran et al., 2019) (Figure 6), and a large organogenesis dataset with $n = 2$ million cells (Cao et al., 2019) (Figure 7). For smaller datasets, the resulting embeddings were mostly similar, suggesting comparable performance between methods. Hafemeister and Satija (2019) argued that using Pearson residuals reduces the amount of variance in the embedding explained by the sequencing depth variation, compared to sequencing depth normalization and log-transformation. We argue that this

effect was mostly due to the large factor that the authors used for rescaling the counts after normalization (Figure S5): large scale factors and/or small pseudocounts (ϵ in $\log(x+\epsilon)$) are known to introduce spurious variation into the distribution of normalized counts (Lun, 2018; Townes et al., 2019). For the PBMC dataset, all three t-SNE embeddings showed similar amount of sequencing depth variation across the embedding space (Figure S4g–i). Performing the embeddings on 1000 genes with the largest Pearson residual variance did not noticeably affect the embedding quality (Figure S4).

However, on closer inspection, embeddings based on Pearson residuals consistently outperformed the other two. For example, while the Pearson residual embeddings clearly separated fine cell types in the full-retina dataset (Macosko et al., 2015), the square-root embedding mixed some of them (we observed the same when using the log-transform). For the same dataset, GLM-PCA embedding did not fully separate some of the biologically distinct cell types. Furthermore, GLM-PCA embeddings often featured Gaussian-shaped blobs with no internal structure (Figure 6), suggesting that some fine manifold structure was lost, possibly due to convergence difficulties.

Embedding the organogenesis dataset (Cao et al., 2019) using Pearson residuals uncovered a strong and surprising batch artifact: hitherto unnoticed, several genes were highly expressed exclusively in small subsets of cells, with each subset coming from a single embryo. These subsets appeared as isolated islands in the t-SNE embedding (Figure 7), allowing us to uncover and remove this batch effect (Figure S6), leading to the final, biologically interpretable embedding (Figure 7). In contrast, embeddings based on log-transform or GLM-PCA did not show this batch artifact at all. GLM-PCA took days to converge (Table 1) and could recover only the coarse structure of the data. Interestingly, the final embedding based on Pearson residuals was broadly similar to the embedding obtained after log-transform and standardization of each gene, as expected given that Pearson residuals stabilize the variance by

construction (Figure 7). Together, these qualitative observations suggest that analytic Pearson residuals can represent small, distinct subpopulations in large datasets better than other methods.

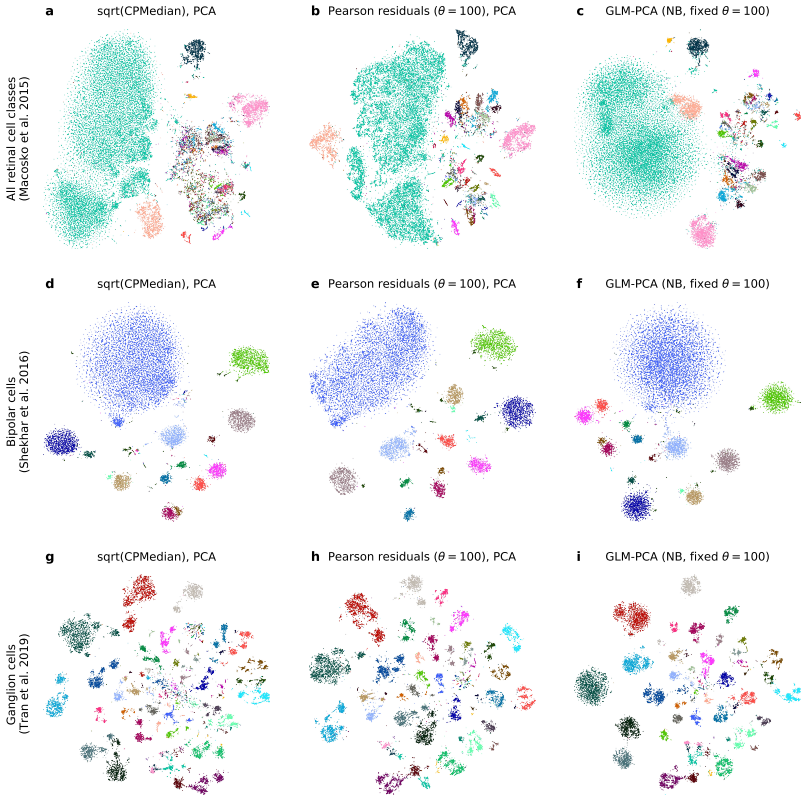

Figure 6: t-SNE embeddings of three retinal datasets. Panels in each column are based on a different data transformation method with PCA or GLM-PCA reduction to 50 dimensions (see Methods), and each row shows a different retinal dataset. We did not perform any gene selection here. Colors correspond to cell type labels provided by the original papers. **a–c:** Full-retina dataset (DropSeq) (Macosko et al., 2015), containing all retinal cell types (including glia and vascular cells). 24 769 cells. **d–f:** Bipolar cell dataset (DropSeq) (Shekhar et al., 2016). 13 987 cells. **g–i:** Retinal ganglion cell dataset (10X v2) (Tran et al., 2019). 15 750 cells.

To quantify the performance of dimensionality reduction methods, we performed a systematic benchmark using the `Zhengmix8eq` dataset

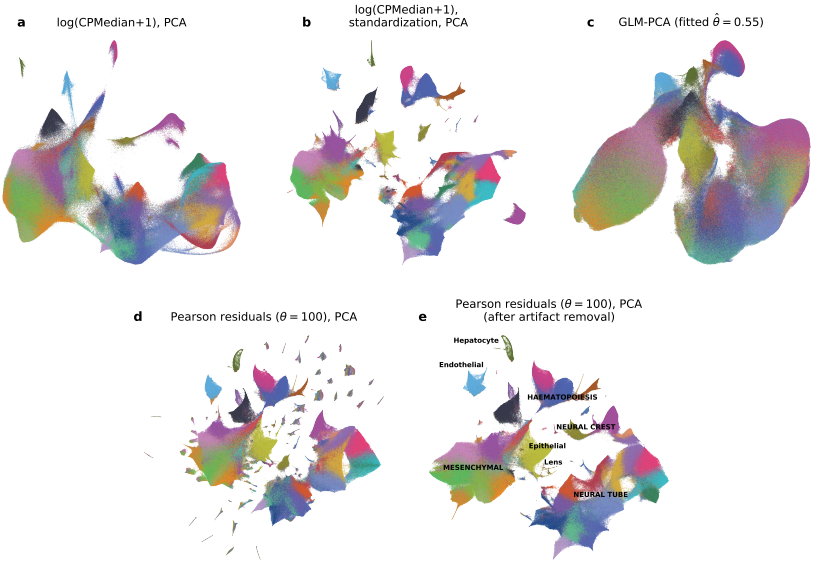

Figure 7: t-SNE embeddings of the organogenesis dataset. All panels show t-SNE embeddings of the organogenesis dataset (Cao et al., 2019) (2 058 652 cells), colored by the 38 main clusters identified by the original authors. All panels use 2 000 genes with the largest Pearson residual variance. Each panel shows a total of 2 026 641 cells, excluding 32 011 putative doublets identified in the original paper. All t-SNE embeddings were done with exaggeration 4 (Kobak and Berens, 2019; Böhm et al., 2020). **a:** Depth-normalization, median scaling, log-transformation and PCA with 50 principal components. **b:** Same as in (a), but with an additional standardization step that scales the normalized and log-transformed expression of each gene to mean zero and unit variance, as in the original paper (Cao et al., 2019). **c:** GLM-PCA with 50 dimensions (NB model with shared overdispersion as a free parameter, estimated to be $\hat{\theta} = 0.56$). **d:** Analytic Pearson residuals with $\theta = 100$ and PCA with 50 principal components. The scattered small islands do not belong to single clusters but instead are spuriously enriched in single embryos. **e:** Same as in (d), but after removing batch-effect genes (Methods). Text labels correspond to the developmental trajectories identified in the original paper (Cao et al., 2019) (uppercase: multi-cluster trajectories, lowercase: single-cluster trajectories).

with known ground truth labels (Duò et al., 2018) (Figure 8). This dataset consists of PBMC cells FACS-sorted into eight different cell types before sequencing (Zheng et al., 2017), with eight types occurring in roughly equal proportions. To make the setup more challenging, we added 10 pseudo-genes expressed only in a group of 50 cells, effectively creating a ninth, rare, cell type (see Methods). We used six methods to select 2000 HVGs (and additionally omitted HVG selection) and ten methods for data transformation and dimensionality reduction to 50 dimensions. We assessed the resulting $(6 + 1) \cdot 10 = 70$ pipelines using kNN classification of cell types. We used the macro F1 score (harmonic mean between precision and recall, averaged across classes) because this metric fairly averages classifier performance across classes of unequal size. Together, the F1 score of the kNN classifier quantifies how well each pipeline separated cell types in the 50-dimensional representation (Figure 8c). We did not include approaches that use depth normalization with inferred size factors (Lun et al., 2016) in this comparison.

The pipeline that used analytic Pearson residuals for both gene selection and data transformation outperformed all other pipelines with respect to cell type classification performance. In contrast, popular methods for HVG selection (e.g., `seurat_v3` as implemented in `Scanpy` (Wolf et al., 2018; Stuart et al., 2019)) combined with log or square-root transformations after depth normalization performed worse and in particular were often unable to separate the rare cell type (Figure 8a,b; see Figure S7 for additional embeddings). The performance of GLM-PCA was also poor, likely due to convergence issues (with 15-dimensional, and not 50-dimensional, output spaces, GLM-PCA performed on par with Pearson residuals; data not shown), in agreement with what we reported above for the retinal datasets. Finally, deviance residuals (Townes et al., 2019) were clearly outperformed by Pearson residuals both as gene selection criterion and as data transformation. This is due to the reduced sensitivity of deviance residuals to low- or medium-expression genes (Figure S2). Note that in terms of the overall classification accuracy no

pipeline outperformed Pearson residuals but many pipelines performed similarly well; this is because overall accuracy is not sensitive to the rare cell type, unlike the macro F1 score.

For this dataset, not using gene selection at all performed similarly well to HVG selection using Pearson residuals (Figure 8c), but in general HVG selection is a recommended step in scRNA-seq data analysis (Luecken and Theis, 2019; Amezquita et al., 2020) and here Pearson

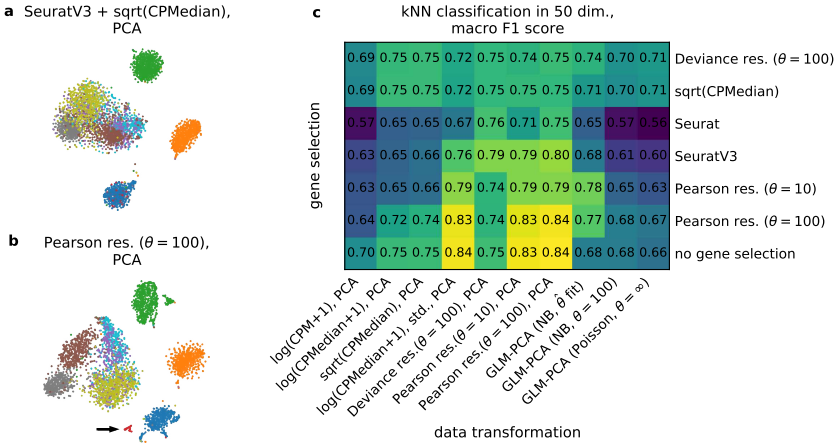

Figure 8: Benchmarking the effect of normalization on cell type separation in reduced dimensionality. We used the `Zhengmix8eq` dataset with eight ground truth FACS-sorted cell types (Zheng et al., 2017; Duò et al., 2018) (3 994 cells) and added ten pseudo-genes expressed in a random group of 50 cells from one type. All HVG selection methods were set up to select 2 000 genes, and all normalization and dimensionality reduction methods reduced the data to 50 dimensions. For details see Methods. **a:** t-SNE embedding after the `seurat_v3` HVG selection as implemented in `Scanpy`, followed by depth-normalization, median scaling, square-root transform, and PCA. Colors denote ground truth cell types, the artificially added type is shown in red. **b:** t-SNE embedding after HVG selection by Pearson residuals ($\theta = 100$), followed by transformation to Pearson residuals ($\theta = 100$), and PCA. Black arrow points at the artificially added type. **c:** Macro F1 score (harmonic mean between precision and recall, averaged across classes to counteract class imbalance) for kNN classification ($k = 15$) of nine ground truth cell types for each of the 70 combinations of HVG selection and data transformation approaches.

residuals performed the best. Also, log-transformed counts that were standardized performed similarly well to Pearson residuals (Figure 8c), in agreement with the above observations on the organogenesis dataset (Figure 7b). Nevertheless, the same organogenesis example showed that Pearson residuals can be more sensitive (Figure 7b, d).

3.2.8 Analytic Pearson residuals are fast to compute

The studied normalization pipelines differ in both space and time complexity. UMI count data are typically very sparse (e.g., in the PBMC dataset, 95% of entries are zero) and can be efficiently stored as a sparse matrix object. Sequencing depth normalization and square-root or log-transformation do not affect the zeros, preserving the sparsity of the matrix, and PCA can be run directly on a sparse matrix. In contrast, Pearson residuals form a dense matrix without any zeros, and so can take a large amount of memory to store (4.5 Gb for the PBMC dataset). For large datasets this can become prohibitive (but note that a smart implementation may be able to avoid storing a dense matrix in memory (Irizarry, 2021)). In contrast, GLM-PCA can be run directly on a sparse matrix but takes a long time to converge (Table 1), becoming prohibitively slow for bigger datasets.

Computational complexity can be greatly reduced if gene selection is performed in advance. After selecting 1000 genes, Pearson residuals do not require a lot of memory (0.3 Gb for the PBMC dataset) and so can be conveniently used. Note that the Pearson residual variance can be computed per gene, without storing the entire residual matrix in memory. GLM-PCA, however, remained slow even after gene selection (4 h vs. 4 s for Pearson residuals for the PBMC dataset; 2 days vs. 4 m for the organogenesis dataset; Table 1).

3.3 Chapter Discussion

We reviewed and contrasted different methods for normalization of UMI count data. We showed that without post-hoc smoothing, the negative

binomial regression model of Hafemeister and Satija (2019) exhibits high variance in its parameter estimates because it is overspecified, which is why it had to be smoothed in the first place. We argued that instead of smoothing an overspecified model, one should resort to a more parsimonious and theoretically motivated model specification involving an offset term. This made the model equivalent to the rank-one GLM-PCA of Townes et al. (2019) and yielded a simple analytic solution, closely related to *correspondence analysis* (Greenacre, 2007). Further, we showed that the estimates of per-gene overdispersion parameter θ_g in the original paper exhibit substantial and systematic bias. . We used negative control datasets from different experimental protocols to show that UMI counts have low overdispersion and technical variation is well described by $\theta \approx 100$ shared across genes.

We found that the approach developed by Hafemeister and Satija (2019) and implemented in the R package `scTransform` in practice yields Pearson residuals that are often similar to our analytic Pearson residuals with fixed overdispersion parameter (Figure S2). We argue that our model with its analytic solution is attractive for reasons of parsimony, theoretical simplicity, and computational speed. Moreover, it provides an explanation for the linear trends in the smoothed estimates in the original paper. We have integrated Pearson residuals into upcoming `Scanpy` 1.9 (Wolf et al., 2018).

Following our manuscript, `scTransform` was updated to `scTransform v2` and now uses the offset model formulation (Choudhary and Satija, 2021). At the same time, the authors argue that the dependence of the overdispersion parameter θ_g on the gene expression strength is not entirely explained by the estimation bias. To reduce the bias, `scTransform v2` uses `glmGamPoi` (Ahlmann-Eltze and Huber, 2020) to estimate the offsets β_{0g} and the overdispersion parameters θ_g (which are then smoothed). The authors also refer to the bulk RNA-seq literature, where it has been observed that the overdispersion parameter grows monotonically with gene expression (Anders and Huber, 2010; Law et al., 2014;

Love et al., 2014). Given the difficulties with estimating overdispersion for low expression means (see above), we believe that this question requires further investigation. However, as argued above, whether θ is assumed to be constant or is allowed to vary between genes, has very little effect on the resulting Pearson residuals.

A parallel publication (Breda et al., 2021) suggested a Bayesian procedure named `Sanity` for estimating expression strength underlying the observed UMI counts, based on Poisson likelihood and Bayesian shrinkage. Importantly, Pearson residuals are not aiming at estimating the underlying expression strength; rather, they quantify how strongly each observed UMI count deviates from the null model of constant expression across cells. These two approaches can have opposite effects on gene markers of rare cell types: the Bayesian procedure shrinks their expression towards zero whereas our approach yields large Pearson residuals. We argued here that this emphasis on rare cell types is useful for many downstream tasks, but if the interest lies in true expression, approaches like `Sanity` may be more appropriate. Future work should perform comprehensive benchmarks on a variety of tasks (Ahlmann-Eltze and Huber, 2023).

On the practical side, we showed that Pearson residuals outperform other methods for selecting biologically variable genes. They are also better than other preprocessing methods for downstream analysis: in a systematic benchmarking effort, we demonstrated that Pearson residuals provide a good basis for general-purpose dimensionality reduction and for constructing 2D embeddings of single-cell UMI data. In particular, they are well suited for identifying rare cell types and their genetic markers. Applying gene selection prior to dimensionality reduction reduces the computational cost of using Pearson residuals down to negligible. We conclude that analytic Pearson residuals provide a theory-based, fast, and convenient method for normalization of UMI datasets.

	Sqrt(CPMedian) + PCA	Pearson residuals + PCA	GLM-PCA
33k PBMC	31 s	35 s	15 h
33k PBMC 1000 HVGs	3 s	4 s	4 h
2M organogenesis 2000 HVGs	166 s	224 s	52 h

Table 1: Runtimes for different normalization pipelines. The datasets are: the full 33k PBMC dataset, the PBMC dataset after selecting 1 000 HVGs, and the organogenesis dataset (Cao et al., 2019) after selecting 2 000 HVGs. Genes with largest Pearson residual variances were selected, which took 9 seconds (PBMC) and 15 minutes (organogenesis), respectively. See Methods for details. All runtimes measured on a machine with 256 Gb RAM and 30 CPU threads at 2.1 GHz.

3.4 Chapter Methods

Code and data availability All software needed to reproduce the analysis and figures presented in this work are published under GNU Affero General Public License v3.0 on Github at `https://www.github.com/berenslab/umi-normalization` (Lause, 2021). The state of the repository at submission of this manuscript is archived at `https://zenodo.org/record/5150534`.

All datasets used in this work are publicly available and listed in the following Table 2. Detailed download instructions can be found at our Github repository.

3.4.1 Mathematical details

Analytic solution The log-likelihood for the model defined in Eqs. 11–12

$$X_{cg} \sim \text{Poisson}(n_c p_g) \tag{23}$$

can be, up to a constant, written as

$$\mathcal{L} = \sum_{cg} \Big[X_{cg} \ln(n_c p_g) - n_c p_g \Big], \tag{24}$$

where we used the Poisson density $p(x) = e^x e^{-\mu}/x!$. Taking partial derivatives with respect to n_c and p_g and setting them to zero, one obtains

$$\hat{n}_c = \frac{\sum_g X_{cg}}{\sum_g \hat{p}_g}, \quad \hat{p}_g = \frac{\sum_c X_{cg}}{\sum_c \hat{n}_c}. \tag{25}$$

This is a family of solutions. Setting $\sum_g \hat{p}_g = 1$, we obtain Eq. 13 and the formulas for $\hat{n}_c$ and $\hat{p}_g$ given in Section 2.1.

This derivation does not generalize for the negative binomial model with density

$$p(x) = \frac{\Gamma(x+\theta)}{x!\,\Gamma(\theta)} \Big(\frac{\theta}{\theta+\mu}\Big)^{\theta} \Big(\frac{\mu}{\theta+\mu}\Big)^{x}, \tag{26}$$

where the log-likelihood (for fixed θ), up to a constant, is

$$\mathcal{L} = \sum_{cg} \Big[X_{cg} \ln(n_c p_g) - (X_{cg} + \theta) \ln(n_c p_g + \theta) \Big]. \tag{27}$$

This does not have an analytic maximum likelihood solution. However, for large θ values Eq. 13 can be taken as an approximate solution.

Deviance residuals Deviance is defined as the doubled difference between the log-likelihood of the saturated model and the log-likelihood of the actual model. The saturated model, in our case, is a full rank model with $\hat{\mu}^*_{cg} = X_{cg}$. For the Poisson model, the deviance can therefore be obtained from Eq. 24 and is equal to

$$\mathcal{D} = 2 \sum_{cg} \Big[X_{cg} \ln \frac{X_{cg}}{\hat{\mu}_{cg}} - (X_{cg} - \hat{\mu}_{cg}) \Big], \tag{28}$$

where the terms with $\hat{\mu}_{cg} = X_{cg}$ are taken to be zero.

Deviance residuals are defined as square roots of the respective deviance terms, such that the sum of squared deviance residuals is equal to the deviance (note that for the Gaussian case this already holds true for the raw residuals, because the saturated model has zero log-likelihood, and the deviance is simply the squared error). It follows that for the

Poisson model deviance residuals (Townes et al., 2019) are given by

$$Z_{cg} = \text{sign}(X_{cg} - \hat{\mu}_{cg})\sqrt{2\Big[X_{cg}\ln\frac{X_{cg}}{\hat{\mu}_{cg}} - (X_{cg} - \hat{\mu}_{cg})\Big]} \tag{29}$$

Similarly, for the negative binomial model with fixed θ, the deviance residuals follow from Eq. 27 and are given by

$$Z_{cg} = \text{sign}(X_{cg} - \hat{\mu}_{cg})\sqrt{2\Big[X_{cg}\ln\frac{X_{cg}}{\hat{\mu}_{cg}} - (X_{cg} + \theta)\ln\frac{X_{cg} + \theta}{\hat{\mu}_{cg} + \theta}\Big]} \tag{30}$$

It is easy to verify that this formula reduces to the Poisson case when $\theta \to \infty$. When computing deviance residuals, we estimated $\hat{\mu}_{cg}$ using Eq. 13.

Clipping Pearson residuals Clipping Pearson residuals to $\pm\sqrt{n}$ as suggested by Hafemeister and Satija (2019) is needed to avoid large residual variance in rarely expressed genes (Figure S2d). The intuition behind this heuristic is as follows. Consider a UMI dataset with n cells containing a biologically distinct rare population P of size $m \ll n$. Let this population have a marker gene with expression following Poisson(λ) for the cells from P, and zero expression for all $n - m$ remaining cells. For simplicity we assume the Poisson model here, and further assume that all cells have the same sequencing depth.

The expected average expression of this gene is $\lambda m/n$ and so the expected Pearson residual value for this gene for the cells from P is $(\lambda - \lambda m/n)/\sqrt{\lambda m/n} = (n-m)\sqrt{\lambda/(nm)} \approx \sqrt{\lambda n/m}$.

With the clipping threshold $\sqrt{n}$, clipping will happen whenever $\lambda > m$, i.e., when the population P is either very small or has very large UMI counts. For example, a population of 10 cells having a marker gene with the within-population mean expression of 20 UMIs, will result in clipped residuals, as if the within-population mean expression were ∼10 UMIs. This may have a large effect on the leading principal components (even PC1) if the data contain a very small number of cells with strong marker

gene expression.

Pearson residuals of biologically variable genes It is instructive to observe the effect Pearson residuals have on genes that have the same variance of log-expression but different expression means. Consider a gene that has expression μ in half of the cells and is upregulated by a factor of two in the other half of the cells. Then its expression mean is 1.5μ, and the Pearson residuals are close to $\pm 0.5\mu/\sqrt{1.5\mu} \approx 0.4\sqrt{\mu}$, i.e., the variance of Pearson residuals grows linearly with μ. This makes sense because for higher-expressed genes there is more statistical certainty about over-Poisson variability, but at the same time highlights that Pearson residuals do not aim to estimate the underlying (log-)expression, unlike e.g., `Sanity` (Breda et al., 2021).

3.4.2 Experimental details

Analyzed datasets and preprocessing Used datasets are listed in Table 2. For the organogenesis dataset and the FACS-sorted PBMC dataset, we applied no further filtering. In all remaining datasets we excluded genes that were expressed in fewer than 5 cells, following Hafemeister and Satija (2019). The data were downloaded following links in original publications in form of UMI count tables. Direct links to all data sources are given in our Github repository `https://github.com/berenslab/umi-normalization`.

HVG selection For gene selection using `sqrt(CPMedian)`, Pearson residuals, and deviance residuals, we applied the respective data transformation and used the variance after transformation as selection criterion. For `Seurat` and `Seurat_v3` methods, we used the respective `Scanpy` implementations. In brief, these two methods regress out the mean-variance relationship, and return an estimate of the 'excess' variance for each gene (Satija et al., 2015; Stuart et al., 2019). For `scTransform` we used the corresponding R package (Hafemeister and Satija, 2019). The

Fano factor was computed after normalizing by sequencing depth and scaling by median sequencing depth.

Data transformation and dimensionality reduction We used the following abbreviations to denote data transformations: `sqrt(CPMedian)` — normalization by sequencing depth, followed by scaling by the median depth across all cells ('counts per median'), followed by the square-root transform; `log(CPMedian + 1)` — normalization by sequencing depth, followed by scaling by the median depth across all cells, followed by $\log(x+1)$ transform; `log(CPMedian + 1) + standardization` — same as `log(CPMedian + 1)`, but followed by centering each gene at mean zero and unit variance; `log(CPM + 1)` — normalization by sequencing depth, followed by scaling by one million ('counts per million'), followed by $\log(x+1)$ transform. Pearson residuals were computed with Eq. 22 and then clipped to $\pm\sqrt{n}$. Deviance residuals were computed with Eq. 30.

All of these methods were typically followed by dimensionality reduction by PCA to 50 dimensions using the `Scanpy` implementation (Wolf et al., 2018), unless otherwise stated.

Further, we used three variants of GLM-PCA to transform raw counts and reduce dimensionality down to 50 in a joint step: Poisson GLM-PCA, negative binomial GLM-PCA with estimation of single overdispersion parameter θ shared across genes, and negative binomial GLM-PCA with fixed shared θ. In Townes et al. (2019), the authors only used the former two methods. Whenever possible, we used the `glmpca-py` implementation with default settings. When we reduced the PBMC dataset to 1 000 genes for Figure S4f, GLM-PCA did not converge with default penalty 1, so we increased it to 5, following the tuning procedure used in the authors' `R` implementation. Similarly, negative binomial GLM-PCA with estimation of θ did not converge on the benchmark dataset (Figure 8) when we used gene selection by either Deviance residuals ($\theta = 100$) or Pearson residuals ($\theta = 10$). For these two cases, we had to increase the penalty to 10. On the organogenesis dataset, the Python implementation did not converge within reasonable time, so for this dataset,

we resorted to the R implementation. It uses a different optimization method and employs stochastic minibatches. All reported GLM-PCA results for this dataset are for batchsize 10 000 as batchsizes 100 and 1 000 (default) resulted in considerably longer runtimes. Because the R implementation does not support NB GLM-PCA with fixed theta, for this dataset we used GLM-PCA with jointly fit $\hat{\theta}$.

Unless otherwise stated, all residuals and GLM-PCA with fixed θ used $\theta = 100$. Whenever gene selection was performed prior to a data transformation that required sequencing depths, we computed those depths using the sum over selected genes only.

Benchmarking cell type separation with kNN classification We used the `Zhengmix8eq` dataset with known ground truth labels obtained by FACS-sorting of eight PBMC cell types (Zheng et al., 2017; Duò et al., 2018). There were 400–600 cells in each cell type. We created a ninth, artificial population from 50 randomly selected B-cells (marked blue in Figure 8). To mimic a separate cell type, we added 10 pseudo marker that had zero expression everywhere apart from those 50 cells. For those 50 cells, UMI values were simulated as Poisson($n_i p$), where n_i is the sequencing depth of the i-th selected cell (range: 452–9697), and expression fraction p was set to 0.001.

We then applied the 70 normalization pipelines shown in Figure 8 to this dataset. Each pipeline either used one of the six methods to select 2000 HVGs or proceeded without HVG selection, followed by one of the ten methods for data transformation and dimensionality reduction to 50 dimensions. To assess cell type separation in this output space, we used a kNN classifier with a leave-one-out cross-validation procedure: For each cell, we trained a kNN classifier on the remaining $n-1$ cells. This resulted in a class prediction for each cell based on the majority vote of its $k = 15$ neighboring cells. We quantified the performance of this prediction by computing the macro F1 score (harmonic mean between precision and recall, averaged across classes to counteract class imbalance). We used the `sklearn` implementations for kNN classification and the F1

score (Pedregosa et al., 2011).

Measuring runtimes All runtimes given in Table 1 are wall times from running the code in a Docker container with an Ubuntu 18 system on a machine with 256 GB RAM and 2×24 CPUs at 2.1 Ghz (Xeon Silver 4116 Dodecacore). The Docker container was restricted to use at most 30 CPU threads. To reduce overhead, we did not use `Scanpy` for timing experiments, and instead used `numpy` for basic computations and `sklearn` for PCA with default settings. Note that we used different implementations of GLM-PCA for the PBMC and organogenesis dataset (see above for details).

t-SNE embeddings All t-SNE embeddings were made following recommendations from a recent paper(Kobak and Berens, 2019) using the FIt-SNE implementation (Linderman et al., 2019). We used the PCA (or, when applicable, GLM-PCA) representation of the data as input. We used default FIt-SNE parameters, including automatically chosen learning rate. For initialization, we used the first two principal components of the data, scaled such that PC1 had standard deviation 0.0001 (as is default in FIt-SNE). The initialization was shared among all embeddings shown in the same figure, i.e., PCs of one data representation were used to initialize all other embeddings as well. For all datasets apart from the organogenesis one, we used perplexity combination of 30 and $n/100$, where n is the sample size (Kobak and Berens, 2019). For the organogenesis dataset embeddings we used perplexity 30 and exaggeration 4 (Böhm et al., 2020).

Chapter Acknowledgements

We thank Christoph Hafemeister, Rahul Satija, William Townes, Florian Wagner, Constantin Ahlmann-Eltze, and Erik van Nimwegen for discussions and helpful comments, and the `Scanpy` team for support.

reference	accession no.	protocol	cells	genes	species	description
33k PBMC	*	10X v1	33 148	16 809	human	peripheral blood mononuclear cells
(Zheng et al., 2017)	SRP073767**	10X v1	3 994	15 715	human	FACS-sorted PBMC cells
(Svensson et al., 2017)	E-MTAB-5480	10X v2	2 000	13 025	human	droplets of bulk RNA solution
(Klein et al., 2015)	GSM1599501	inDrop	953	25 025	human	droplets of bulk RNA solution
(Han et al., 2018)	GSM2906413	MicrowellSeq	9 994	15 069	mouse	non-differentiating stem cells
(Macosko et al., 2015)	GSE63472	DropSeq	24 769	17 973	mouse	retinal cells
(Shekhar et al., 2016)	GSE81904	DropSeq	13 987	16 520	mouse	retinal bipolar cells
(Tran et al., 2019)	GSE133382	10X v2	15 750	17 685	mouse	retinal ganglion cells
(Cao et al., 2019)	GSE119945	sci-RNA-seq3	2 058 652	26 183	mouse	organogenesis of mouse embryo cells

Table 2: Overview of UMI datasets used for analysis. In the 10X control dataset (Svensson et al., 2017), we used only sample 1. In the MicrowellSeq control dataset (Han et al., 2018), we used the E14 dataset. In the three retinal datasets (Tran et al., 2019; Shekhar et al., 2016; Macosko et al., 2015), we only used cells from the largest batch. The FACS-sorted PBMC dataset was assembled by authors of a recent paper(Duò et al., 2018), based on a benchmarking dataset published earlier (Zheng et al., 2017). Numbers of genes and cells are after batch selection (where applicable) and initial gene filtering (see Methods). Scripts performing these operations and detailed download instructions for all materials are published in our Github repository at `http://www.github.com/berenslab/umi-normalization`. The accession numbers refer to archived datasets at the Gene Expression Omnibus (NCBI), the Sequence Read Archive (NCBI), or ArrayExpress (EMBL-EBI). (*): Data directly obtained from 10X Genomics at `https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc33k`. (**): The accession number links to the base dataset that the original authors used to construct the ground-truth dataset for their paper (Duò et al., 2018). To obtain the dataset used here, use the `Bioconductor 3.1.3` R package `DuoClustering2018` or visit the authors' website (`http://imlspenticton.uzh.ch/robinson_lab/DuoClustering2018/`).

4 Compound models for non-UMI counts

Publication note. This chapter is published as a preprint on *bioRxiv* (Lause et al., 2023) and is available at `https://doi.org/10.1101/2023.08.02.551637` under a `CC-BY-NC-ND 4.0` license (`https://creativecommons.org/licenses/by-nc-nd/4.0/`). At the time of writing, a revised version is under review at *Genome Biology*.

Abstract

Recent work employed Pearson residuals from Poisson or negative binomial models to normalize UMI data. To extend this approach to non-UMI data, we model the additional amplification step with a compound distribution: we assume that sequenced RNA molecules follow a negative binomial distribution, and are then replicated following an amplification distribution. We show how this model leads to compound Pearson residuals, which yield meaningful gene selection and embeddings of Smart-seq2 datasets. Further, we suggest that amplification distributions across several sequencing protocols can be described by a broken power law. The resulting compound model captures previously unexplained overdispersion and zero-inflation patterns in non-UMI data.

4.1 Introduction

Single-cell RNA sequencing (scRNA-seq) data are affected by count noise and technical variability due to the total number of sequenced molecules varying from cell to cell. Removing this technical variation by normalization and variance stabilization is an important step in common analysis pipelines (Luecken and Theis, 2019; Heumos et al., 2023). The standard approach for this has been to use the $\log(1 + x/s)$ transformation, where s is a size factor of the cell. While the log-transform often performs well in practice (Ahlmann-Eltze and Huber, 2023), it has well-known theoretical limitations and can produce biased results (Lun, 2018).

Recently, a number of count modelling approaches like `sctransform` (Hafemeister and Satija, 2019), GLM-PCA (Townes et al., 2019),

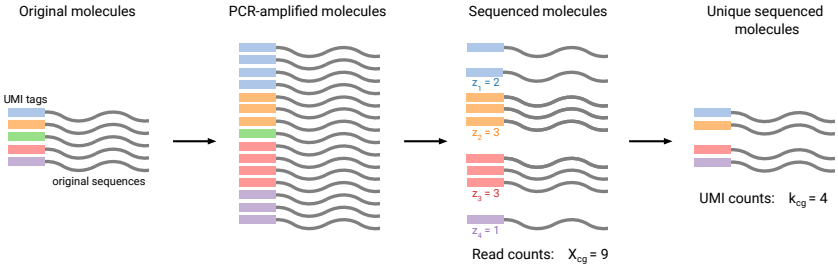

Figure 9: Important quantities in single-cell RNA sequencing. A cell c contains some number (here, 5) of RNA molecules of gene g. In UMI-based protocols, the original molecules are tagged by unique molecular identifiers (UMIs) before PCR amplification and sequencing. Both processes are imperfect, such that not all original molecules get amplified by the same factor, and not all amplified molecules get sequenced. In the end, each of the unique molecules may get sequenced one or several times, which is its *copy number* (z_i). Here z_i values are 2, 3, 3, and 1, and one original molecule (green) is not detected at all. The sum of all copy numbers gives the observed *read count* X_{cg} (here, 9). UMIs allow to compute a *UMI count* k_{cg} by only counting unique sequenced molecules (here, 4). In non-UMI protocols, amplification and sequencing work the same, but the final de-duplication step is not possible, meaning that only X_{cg} is observable. Note that the shown numbers are not to scale: A typical cell might contain on the order of 10^5 RNA molecules across all genes (Ziegenhain et al., 2022); yielding on the order of 10^{10} molecules after amplification; and producing on the order of 10^6 sequenced reads.

Sanity (Breda et al., 2021), and analytic Pearson residuals (Lause et al., 2021) have been suggested for preprocessing scRNA-seq data. These methods are based on explicit statistical models of the count generation process, rather than on heuristics such as the log-transform. One limitation of all of these methods is that they are tailored to data obtained using sequencing protocols based on unique molecular identifiers (UMIs), and are not appropriate for non-UMI technologies such as Smart-seq2 (Picelli et al., 2013). In this paper, we develop a count model and corresponding analytic Pearson residuals (Lause et al., 2021) for non-UMI sequencing data.

Single-cell sequencing protocols usually require an amplification step by polymerase chain reaction (PCR) to obtain enough starting mate-

rial for sequencing (Figure 9). The process is imperfect, and different molecules will get amplified to a different extent. As a result, the number of sequenced molecules of a given gene (called *read count*) does not reflect the original number of RNA molecules in the cell.

In UMI protocols, a random DNA sequence called UMI is appended to each original reverse-transcribed RNA molecule prior to the amplification, and is then amplified and sequenced together with it (Islam et al., 2014). Because the UMI uniquely identifies each original molecule, one can later remove amplification duplicates by counting each UMI only once ('de-duplication'), giving rise to the *UMI counts* instead of *read counts* (Figure 9), effectively reducing amplification noise (Grün et al., 2014). Note that the UMI count does not necessarily equal the number of original RNA molecules present in the cell, as some molecules can get lost during sample preparation for the sequencing ('library preparation') or fail to get sequenced due to low capture rate ('depth') in the sequencing step (Figure 9).

While UMI protocols are popular in the scRNA-seq community, non-UMI technologies remain important. Indeed, UMIs only mark one end of the original molecules, so UMI counts are not available for the internal reads (most protocols involve a fragmentation step that cuts each molecule into small fragments prior to sequencing). Full-length sequencing methods, such as Smart-seq2 (Picelli et al., 2013), are typically more sensitive than UMI protocols (Ziegenhain et al., 2017; Ding et al., 2020), and are often used to detect rare cell types (Tasic et al., 2018; Yao et al., 2021) or splicing variants (Feng et al., 2021), or in low-throughput experiments such as Patch-seq (Lipovsek et al., 2021). In the resulting datasets, UMI counts are not available, and computational analysis has to be based on the read counts that still contain amplification-induced variability.

Only few normalization methods have been developed specifically to account for the amplification noise in read counts. The *Census* (Qiu et al., 2017) and *quasi-UMIs* (Townes and Irizarry, 2020) methods are

two transformations that are designed to make the shape of the read count distribution approximately match the shape of the UMI count distribution. Afterwards, the transformed data still requires a UMI-like normalization. However, neither Census nor quasi-UMIs derive their transforms from a principled statistical model and rather rely on heuristics.

Here, we develop a new theoretically motivated method for normalization of non-UMI data that explicitly accounts for the amplification noise: *compound Pearson residuals*. We do so by extending the null model behind the analytic Pearson residuals (Lause et al., 2021) from UMI counts to read counts, based on an explicit statistical model for the amplification step. This yields a generative model for read counts that reproduces characteristic patterns of non-UMI data. We demonstrate that our compound Pearson residuals can efficiently normalize complex read count datasets.

4.2 Results

4.2.1 Analytic Pearson residuals for normalization of UMI data

In this section we briefly summarize the normalization approach based on Pearson residuals, originally developed by Hafemeister and Satija (2019) for UMI data. Pearson residuals compare the observed data to a null model that captures only technical variability due to count noise and variations in sequencing depth. The null model assumes perfect biological homogeneity, and so any deviation from it suggests biological variability.

Under the null model (Lause et al., 2021), a gene g takes up a certain constant fraction p_g of the total n_c RNA molecules sequenced in cell c, and the observed UMI counts k_{cg} follow a negative binomial (NB)

distribution:

$$k_{cg} \sim \mathrm{NB}(\mu_{cg}, \theta), \tag{31}$$

$$\mu_{cg} = n_c p_g, \tag{32}$$

where θ is the inverse overdispersion parameter. Higher values of θ yield smaller variance, and for $\theta = \infty$, the NB distribution reduces to the Poisson distribution. Note that θ in this formulation is shared between all genes; based on negative control UMI data, Lause et al. (2021) argued that θ can be set to $\theta = 100$, which is close to Poisson. Sarkar and Stephens (2021) even suggest pure Poisson as measurement model.

Given an observed UMI count matrix, the maximum likelihood estimate of μ_{cg} is given by:

$$\hat{\mu}_{cg} = \frac{\sum_j k_{cj} \cdot \sum_i k_{ig}}{\sum_{ij} k_{ij}}, \tag{33}$$

which is exact in the Poisson case and holds only approximately in the NB case (Lause et al., 2021). This yields the analytic formula for *UMI Pearson residuals* (difference between observed UMI count values and model prediction, divided by the model standard deviation):

$$R_{cg}^{\mathrm{UMI}} = \frac{k_{cg} - \hat{\mu}_{cg}}{\sqrt{\hat{\mu}_{cg} + \hat{\mu}_{cg}^2/\theta}}, \tag{34}$$

where $\hat{\mu}_{cg} + \hat{\mu}_{cg}^2/\theta$ is the variance of the NB distribution with mean $\hat{\mu}_{cg}$ and overdispersion parameter θ. The variance of Pearson residuals does not depend on p_g, and in a homogeneous dataset is close to 1 for all genes. This ensures variance stabilization across all levels of gene expression.

This algorithm is similar to the one implemented in `sctransform` (Hafemeister and Satija, 2019; Choudhary and Satija, 2021) and is equivalent to a rank-one GLM-PCA (Townes et al., 2019), but it is simpler and faster to compute than either of these methods. When followed by

singular value decomposition (SVD), the Poisson ($\theta = \infty$) version of UMI Pearson residuals is also known as correspondence analysis (Hsu and Culhane, 2023).

4.2.2 Compound Pearson residuals for non-UMI read count data

To apply Pearson residuals to scRNA-seq data without UMIs, we need to change the null model, because read counts do not follow the NB distribution (Svensson, 2020; Cao et al., 2021). As in the UMI case, we assume that the number of unique sequenced RNA molecules k_{cg} follows a Poisson or a NB distribution. However, during the amplification step, each of these k_{cg} unique molecules could have been duplicated multiple times before sequencing (Figure 9). For the i-th unique molecule, we call the number of its sequenced duplicates its *copy number* z_i. We assume that copy numbers follow some distribution Z, which we call *amplification distribution*. Our assumption is that the amplification distribution is the same for all genes and all cells, and only depends on the details of the PCR amplification and the sequencing protocol (see the note below about the variable gene length).

The read count X_{cg} of a given gene g in cell c is thus modeled as the sum of k_{cg} independent and identically distributed (i.i.d.) positive integer copy numbers drawn from Z:

$$X_{cg} = \sum_{i=1}^{k_{cg}} z_i, \tag{35}$$

$$z_i \sim \text{Z with } z_i \in \mathbb{N}^+, \tag{36}$$

$$k_{cg} \sim \text{NB}(\mu_{cg}, \theta), \tag{37}$$

$$\mu_{cg} = n_c p_g. \tag{38}$$

For example, $k_{cg} = 4$ means that four unique RNA molecules of gene g were sequenced in a cell c; if their copy numbers were 2, 3, 3, and 1, this would yield the read count value $X_{cg} = 2 + 3 + 3 + 1 = 9$ (c.f.

Figure 9). The resulting distribution of X_{cg} can be called *compound NB distribution* (see Methods).

In the above formulation, our model does not explicitly account for gene length. In most sequencing protocols, longer transcripts are cut into more fragments before amplification. This will result in more unique sequenced fragments k_{cg} for longer molecules (Phipson et al., 2017). In our model, this increase amounts to a constant length factor l_g per gene, which can be absorbed by our per-gene expression fraction p_g. The variance of Pearson residuals does not depend on p_g (see above), so for simplicity, we do not explicitly put gene lengths into the model.

To obtain Pearson residuals for this null model, we need to obtain expressions for its mean and variance. The mean of a compound NB distribution is equal to the product of the NB mean μ_{cg} and the mean of the amplification distribution Z:

$$\mathbb{E}[X_{cg}] = \mathbb{E}[Z] \cdot \mathbb{E}[k_{cg}] = \mathbb{E}[Z] \cdot \mu_{cg}. \tag{39}$$

We can use the observed read count matrix X to estimate the means of the maximum likely null compound model as follows:

$$\hat{\mathbb{E}}[X_{cg}] \approx \mathbb{E}[Z] \cdot \hat{\mu}_{cg} = \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z]} \cdot \frac{\sum_j k_{cj} \cdot \sum_i k_{ig}}{\sum_{ij} k_{ij}} \approx \frac{\sum_j X_{cj} \cdot \sum_i X_{ig}}{\sum_{ij} X_{ij}}. \tag{40}$$

Note that this expression has the same form as Equation 33: the outer product of the row and the column sums of the count matrix, normalized by its total sum.

The mean-variance relationship of the compound NB distribution takes the form (see Methods):

$$\mathrm{Var}[X_{cg}] = \alpha_Z \mathbb{E}[X_{cg}] + \frac{\mathbb{E}[X_{cg}]^2}{\theta}, \tag{41}$$

$$\text{where } \alpha_Z = \mathbb{E}[Z] + \mathbb{FF}[Z] = \mathbb{E}[Z] + \frac{\mathrm{Var}[Z]}{\mathbb{E}[Z]}. \tag{42}$$

This expression is similar to the mean-variance relationship of the NB distribution but contains a scaling parameter α_Z equal to the sum of the mean and the Fano factor (denoted $\mathbb{FF}[Z]$) of Z. Note that in the compound Poisson case ($\theta = \infty$), the compound variance is proportional to the compound mean.

Using these equations, we can compute the Pearson residuals of the compound NB null model, which we call the *compound Pearson residuals*:

$$R_{cg} = \frac{X_{cg} - \hat{\mathbb{E}}[X_{cg}]}{\sqrt{\alpha_Z \hat{\mathbb{E}}[X_{cg}] + \hat{\mathbb{E}}[X_{cg}]^2/\theta}}. \tag{43}$$

Following the UMI case and the arguments in Lause et al. (2021), we set the overdispersion parameter to $\theta = 100$. The scalar α_Z is a function of the mean and variance of the amplification distribution and remains as the only free parameter of the model. Following Hafemeister and Satija (2019) and Lause et al. (2021), we clip the residuals to $[-\sqrt{n}, \sqrt{n}]$, where n is the number of cells in the dataset.

This formalism naturally generalizes the UMI Pearson residuals. Indeed, in the UMI case, each sequenced molecule is counted only once, thanks to the UMIs, meaning that the Z distribution is a delta peak $\delta(1)$ with $\mathbb{E}[Z] = 1$ and $\mathrm{Var}[Z] = 0$, and hence $\alpha_Z = 1$. In this case, Equation 43 reduces to Equation 34.

Conveniently, compound Pearson residuals are equivalent to UMI Pearson residuals of the read count matrix scaled by $1/\alpha_Z$:

$$R_{cg}(X_{cg}; \alpha_Z, \theta) = \frac{(X_{cg} - \hat{\mathbb{E}}[X_{cg}])/\alpha_Z}{\sqrt{\big(\alpha_Z \hat{\mathbb{E}}[X_{cg}] + \hat{\mathbb{E}}[X_{cg}]^2/\theta\big)/\alpha_Z^2}} = R_{cg}^{\mathrm{UMI}}(X_{cg}/\alpha_Z; \theta). \tag{44}$$

Importantly, the necessary scaling factor is not equal to $\mathbb{E}[Z]$, as could be naïvely expected, but rather to $\alpha_Z = \mathbb{E}[Z] + \mathrm{Var}[Z]/\mathbb{E}[Z]$.

Compound Pearson residuals have the same computational complex-

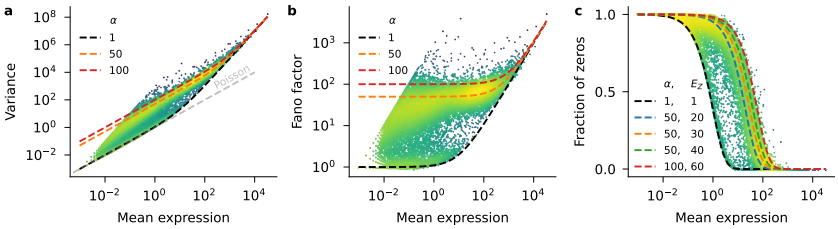

Figure 10: The compound NB model captures statistics of homogeneous read count data. The data are a homogeneous subset of a mouse visual cortex dataset (Tasic et al., 2018) sequenced with Smart-seq2 (*L6 IT VISp Penk Col27a1* cluster; 1 049 cells, 33 914 genes). Each dot represents a gene. Brighter colors indicate higher density of points. Dashed lines show the behavior of the compound negative binomial model ($\theta = 10$). **a:** Mean-variance relationship. Gray line illustrates the Poisson case where mean equals variance. **b:** Relationship between mean expression and Fano factor (variance/mean). **c:** Relationship between mean expression and fraction of zero counts.

ity as UMI Pearson residuals, and can be computed in seconds even for large datasets with >10 000 cells. The matrix of Pearson residuals is dense and will thus require more memory than the sparse matrix of raw counts. For very large datasets it may be prohibitive to hold the full matrix in memory, but memory demand can be reduced by advanced implementations (Irizarry, 2021) or by subsetting to highly variable genes (Lause et al., 2021), allowing to process datasets with millions of cells.

4.2.3 Compound model can fit homogeneous read count data

The compound model introduced above is designed to capture only technical, but not biological variance in non-UMI read count data. Therefore, it should provide a good fit to data that contain little biological variation. To test this, we took scRNA-seq data from adult mouse neocortex sequenced with Smart-seq2 (Tasic et al., 2018), and focused on a subset of cells corresponding to one specific cell type, assuming that there is little biological variability within a cell type. We chose the *L6 IT VISp*

Penk Col27a1 type (as annotated by the authors of the original study), containing 1 049 excitatory neurons (Figure 10).

The mean-variance relationship across genes (Figure 10a) showed that most genes exhibited overdispersion compared to the Poisson model (gray line, $\alpha_Z = 1$ and $\theta = \infty$). Most genes also showed more variance than expected from a NB model without amplification (black line, $\alpha_Z = 1$ and $\theta = 10$). In contrast, compound models accounting for amplification with $\alpha_Z \in [10, 100]$ (colored lines) were able to approximate the mean-variance relationship for the majority of the genes. Note that we used $\theta = 10$ for illustrations in Figure 10 because this value fit the within-cell-type data better than $\theta = 100$, in agreement with the idea that even biologically homogeneous data can show some additional variability on top of the purely technical variability (Lause et al., 2021). Cell-to-cell variation in sequencing depth also contributed to this increase in overdispersion (Supplementary Figure S8).

The relationship between the mean and the Fano factor across genes (Figure 10b) allowed us to further constrain the amplification parameter α_Z. Indeed, for genes with low average expression, the Fano factor of the read counts is approximately equal to α_Z (Equation 41 and Equation 63 in the Methods). The Fano factors of most genes were bounded by models with $\alpha_Z = 1$ from below (black), and by models with $\alpha_Z = 100$ from above (red). The bulk of the genes followed a model with $\alpha_Z = 50$ (orange).

While knowing α_Z is sufficient to compute Pearson residuals, we can obtain separate estimates of $\mathbb{E}[Z]$ and $\mathrm{Var}[Z]$ by studying the relationship between the average expression and the fraction of zeros. This relationship only depends on $\mathbb{E}[Z]$ (see Methods, Equation 64), allowing to estimate this term directly. We observed that in the Smart-Seq2 data, the fraction of observed zeros decreased with increasing mean expression (Figure 10c). There were more observed zeros than expected from a NB model with $\alpha_Z = 1$ (black), hinting at why read count data have in the past often been modeled using a zero-inflated negative binomial (ZINB)

distribution (e.g. Lopez et al., 2018; Chen et al., 2018). Our compound NB model with $\mathbb{E}[Z] \approx 30$ (orange) provided a good qualitative fit to the observed data, without any explicit zero-inflation terms. From here we can compute $\mathbb{FF}[Z] = \alpha_Z - \mathbb{E}[Z] \approx 20$ and hence $\mathrm{Var}[Z] \approx 600$.

The compound NB model with $\alpha_Z = 50$ and $\mathbb{E}[Z] = 30$ described the majority of genes well. However, some genes were instead following the model without any amplification (black line in Figure 10), as if their transcripts were not amplified by the PCR. To understand this pattern, we obtained gene type annotations from `mygene.info`. This revealed that protein-coding genes generally followed our best-fitting compound model (Supplementary Figure S9a–c), while most of the seemingly non-amplified genes were pseudogenes (Supplementary Figure S9d–f). This observation was not limited to the Smart-seq2 data, but also occurred for all sequencing protocols studied in Ziegenhain et al. (2017) (Supplementary Figure S10). Exonic transcript lengths from the `mygene.info` database were shorter for the non-amplified genes (Supplementary Figure S11).

In summary, we showed that our compound NB model fits a biologically homogeneous example dataset. In particular, our model matched the main statistical properties of protein coding genes (mean, variance, and fraction of zeros).

4.2.4 Compound Pearson residuals for normalization of heterogeneous read count data

Next, we computed compound Pearson residuals with $\alpha_Z = 50$ (Equation 43) to preprocess the entire dataset from Tasic et al. (2018), which is highly heterogeneous and includes both neural and non-neural cells from two areas of the mouse neocortex.

For highly variable gene (HVG) selection, we used the variance of compound Pearson residuals for each gene (Figure 11a). Most genes had residual variance close to 1, indicating that they followed the null model. The interpretation is that those genes did not show biological

variability and were not differentially expressed between cell types. In contrast, genes with residual variance $\gg 1$ had more variability than predicted by the null model, implying nontrivial biological variability. For downstream analysis, we selected the 3 000 HVGs with highest residual variances. Among those were well-known marker genes of specific cortical cell types (Figure 11a, red dots). In particular, the four genes with highest residual variance (star symbols in Figure 11a) marked different groups of inhibitory neurons (*Npy*, *Vip*, *Sst*) and astrocytes (*Apoe*). As before, we confirmed that genes with very low residual variances $\ll 1$ were mostly pseudogenes (Supplementary Figure S12).

Next, we used PCA and t-SNE to visualize the single-cell composition of the mouse cortex using compound Pearson residuals of the selected HVGs. The resulting embedding showed rich structure that corresponded well to the cell type annotations originally determined by Tasic et al. (2018) (Figure 11b): individual cell types formed mostly clearly delineated clusters, while related cell types (having similar colors) mostly stayed close to each other. The expression of most variable genes according to the residual variance was typically localized in one part of the embedding space (Figure 11c).

Calculating compound Pearson residuals requires to set the amplification parameter α_Z. To investigate the influence of this parameter, we computed compound Pearson residuals for a range of α_Z values covering three orders of magnitude (Supplementary Figure S13). We found that for $\alpha \gg 1$, the exact value did not lead to large differences in the HVG selection or t-SNE representation. In contrast, when we used UMI Pearson residuals of the null model without amplification ($\alpha_Z = 1$), the HVG selection failed to include some of the most important marker genes (Supplementary Figure S13a) and the embedding quality visibly degraded (Supplementary Figure S13b). This shows that it is not appropriate to apply the original formulation of UMI Pearson residuals (Lause et al., 2021) to non-UMI data, and that it is important to explicitly account for the PCR-induced variance. Reassuringly, the exact value

of the α_Z parameter did not have a large influence on the downstream performance.

We compared our approach to existing methods for read count normalization: qUMI (Townes et al., 2019) and Census (Qiu et al., 2017).

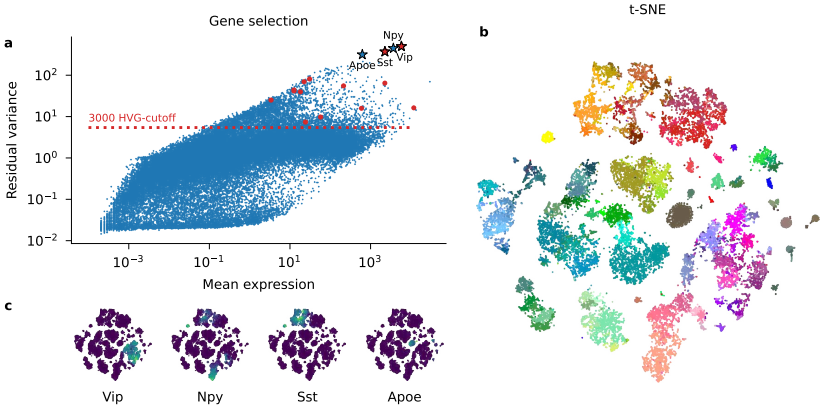

Figure 11: Compound Pearson residuals work well for preprocessing a heterogeneous Smart-seq2 dataset. Here, we used the raw counts of a mouse visual cortex dataset sequenced with Smart-seq2 (Tasic et al., 2018) (23 822 cells, 38 510 genes). We used a compound NB model with amplification parameter $\alpha = 50$ and overdispersion parameter $\theta = 100$. **a:** Highly variable gene (HVG) selection by largest residual variance. Each dot is a gene; genes above the red line were included in the selection of 3 000 HVGs. Stars indicate the top four HVGs shown in panel (c). Red dots and stars correspond to the following well-known marker genes taken from Tasic et al. (2016), from left to right: *Itgam* (microglia), *Bgn* (smooth muscle cells), *Pdgfra* (oligodendrocyte precursors), *Aqp4* (astrocytes), *Flt1* (endothelial cells), *Foxp2* (layer 6 excitatory neurons), *Mog* (oligodendrocytes), *Rorb* (layer 4 excitatory neurons), *Pvalb* (subset of inhibitory neurons), *Slc17a7* (excitatory neurons), *Gad1* (inhibitory neurons), *Sst* (subset of inhibitory neurons), *Vip* (subset of inhibitory neurons), *Snap25* (neurons). **b:** t-SNE embedding on compound Pearson residuals following HVG selection (to 3 000 HVGs) and PCA (down to 1 000 PCs). Each dot is a cell, colored by the original cluster assignments from Tasic et al. (2018). Warm colors: inhibitory neurons. Cold colors: excitatory neurons. Brown and gray colors: non-neural cells. **c:** t-SNE embeddings as in panel (b), colored by expression strength of the four most variable genes according to compound Pearson residual variance. For expression, we show square-root-transformed, depth-normalized counts.

Both use heuristics to estimate UMI counts from read count data, and one can then apply standard UMI methods for further processing. We found that both methods, when combined with UMI Pearson residuals, gave results that were similar to our compound Pearson residuals (Supplementary Figure S14). This is unsurprising, as Census amounts to dividing read counts by a cell-specific constant, and compound Pearson residuals are equivalent to UMI Pearson residuals after appropriate scaling of the data matrix (Equation 44). The qUMI transformation is non-linear but gave similar results for our data. Importantly, both Census and qUMI rely on heuristics (see Discussion), while our approach is based on an explicit statistical model.

We also compared this approach to the default preprocessing implemented in the `Scanpy` library (Wolf et al., 2018) based on depth normalization, `log1p()` transform, and `Seurat` HVG selection. We found that many high-expression genes did not get selected by this method, including known marker genes like *Snap25* (Supplementary Figure S15a). The t-SNE embedding based on the default `Scanpy` preprocessing was similar to ours, but arguably showed less local structure (Supplementary Figure S15b). In the absence of ground truth cell labels, it is impossible to assess the representation quality objectively; however, based on the variance of known marker genes, we argue that compound Pearson residuals provide a more meaningful representation of the data.

As noted above, computing compound Pearson residuals was fast. For this dataset with ca. 23 000 cells and 38 000 genes it took ∼15 s on a single CPU. The resulting dense matrix of residuals used 3.4 Gb of RAM instead of 1.6 Gb for the sparse matrix of read counts. Census and qUMI had slower runtimes (3 h and 3 min respectively).

4.2.5 Compound Pearson residuals recover ground truth

To confirm that compound Pearson residuals are indeed able to recover true marker genes and true cell classes, we simulated read count data with known ground truth based on the Tasic et al. (2018) dataset. In

our simulations, sequencing depths n_s, gene fractions p_g, and class identities were taken from the real data, and we used a compound model with NB($\theta = 100$) as UMI distribution and a geometric distribution as amplification distribution to simulate counts within each class. The true α_Z in this case is equal to 199 (see Methods).

In *Simulation I*, we allowed only a small set of known marker genes to vary between classes. Compound Pearson residuals showed high residual variance only for those ground truth marker genes (even when α_Z was misspecified, Supplementary Figure S16a–b). In contrast, UMI Pearson residuals ($\alpha_Z = 1$) showed high residual variance for many non-variable genes (Supplementary Figure S16c).

In *Simulation II*, we mirrored the full cluster structure of the data by using cluster-specific p_g values for all genes. Using compound Pearson residuals, we obtained reasonable residual variances and embeddings recovering ground truth clusters (regardless of the exact value of $\alpha \gg 1$, Supplementary Figure S16d–e, g–h). At the same time, UMI Pearson residuals failed to stabilize the variance and incorrectly merged many clusters (Supplementary Figure S16f,i). In summary, both simulation experiments confirmed that compound Pearson residuals can recover true marker genes and cell types.

4.2.6 The broken zeta distribution as amplification model

So far, we did not specify the amplification distribution Z. Instead, we only characterized its mean and variance through the α_Z parameter. While this was sufficient to compute compound Pearson residuals, an explicit amplification distribution is needed for the complete specification of the compound model, enabling likelihood calculation or using it as a generative model. To find an appropriate statistical model, we obtained empirical amplification distributions from experimental data generated with several UMI-based protocols: CEL-seq2, Drop-seq, MARS-seq, and SCRB-seq (Ziegenhain et al., 2017), as well as Smart-Seq3 (Hagemann-Jensen et al., 2020) and Smart-Seq3xpress (Hagemann-Jensen et al.,

2022). Together, we analyzed 106.3 million UMIs from 856 cells.

For each UMI barcode, we computed the number of times it occurred in the sequenced reads (its copy number). The normalized histogram of these copy numbers provided an empirical characterization of the amplification distribution Z (Figure 12). Across all sequencing protocols, the copy number histograms in log-log coordinates showed a characteristic elbow shape: Higher copy numbers were less frequent, and the distribution followed two separate decreasing trends in two ranges of copy numbers. This shape can be described by a broken power law, i.e., two separate power laws for low and high copy numbers. The exact shape of the distribution differed between sequencing protocols, leading to different values of mean and variance (Table 3). These values are influenced by how deeply a sample is sequenced and how many cycles of amplification are employed: e.g., the Smart-seq3 Xpress dataset was sequenced

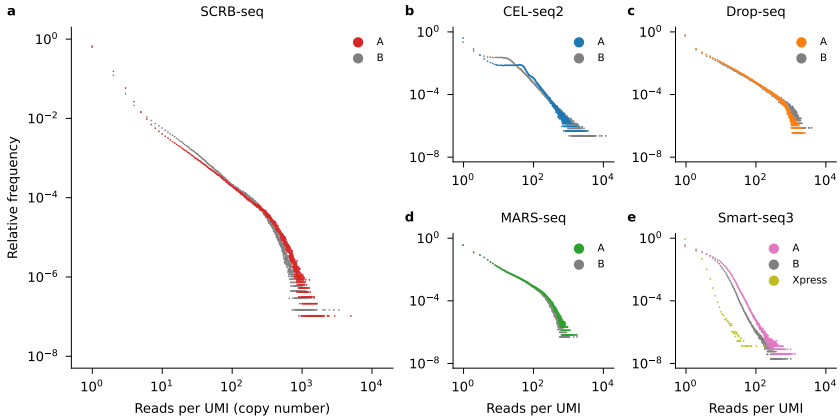

Figure 12: Observed amplification distributions follow a similar shape across protocols. Each panel shows a distribution of UMI copy numbers for a given UMI protocol. **a–d:** SCRB-seq, CEL-seq2, Drop-seq, and MARS-seq data from Ziegenhain et al. (2017). For each protocol two identical runs were performed (A and B). **e:** Smart-seq3 protocols. Data from a single-end experiment (A), a paired-end experiment (B) (Hagemann-Jensen et al., 2020), and a Smart-seq3 Xpress experiment (Hagemann-Jensen et al., 2022).

shallower than the other two Smart-seq3 datasets (ca. 94 000 vs ca. 572 000–806 000 reads per cell) and with fewer PCR cycles, leading to substantially lower $\mathbb{E}[Z]$ values.

To study how stable the amplification distribution was across cells in the same sample, we computed per-cell estimates of $\mathbb{E}[Z]$ and $\alpha_Z = \mathbb{E}[Z] + \mathbb{FF}[Z]$. The estimates showed some variability across cells (Supplementary Figure S17), but it was small enough that our assumption of the shared amplification parameters seems justified in practice. Moreover, the per-cell estimates of α_Z were correlated with the total number of read counts per cell (Supplementary Figure S18), and this between-cell variability is accounted for in our model in any case.

Note that for all protocols, the empirical distribution of copy numbers was monotonically decreasing, meaning that $z = 1$ was the most likely copy number, despite many cycles of amplification. This may seem

Protocol	Run	Cells	UMIs	$\mathbb{E}[Z]$	$\mathbb{FF}[Z]$	α_Z	$\max(z_i)$
CEL-seq2	A	34	2 140 365	27.2	107.5	134.8	3 476
CEL-seq2	B	37	4 303 956	24.4	199.8	224.2	11 092
Drop-seq	A	42	2 506 244	29.6	284.2	313.8	2 463
Drop-seq	B	34	1 272 895	31.1	414.1	445.2	3 718
MARS-seq	A	29	1 342 232	24.1	132.1	156.2	1 624
MARS-seq	B	36	1 903 673	20.9	107.0	128.0	1 719
SCRB-seq	A	39	9 429 371	9.7	218.8	228.5	4 840
SCRB-seq	B	45	6 800 371	9.4	159.3	168.7	3 276
Smart-seq3	A	145	21 549 849	5.4	8.0	13.4	1 216
Smart-seq3	B	319	48 073 893	3.8	4.5	8.3	989
Smart-seq3 Xpress		96	6 940 889	1.3	0.3	1.6	173
Smart-seq2		23 822	—	—	—	—	—

Table 3: Key statistics of the observed amplification distribution across protocols. Top part: Each row in the table corresponds to one of the datasets presented in Figure 12. The number of UMIs shows how many observed copy numbers z_i were used to compute the statistics for that dataset. Bottom row: The Smart-seq2 dataset analyzed in Figure 11, where UMIs were not observed.

counter-intuitive, but previous work (Best et al., 2015) showed that a mechanistic model of the amplification process followed by Poisson sampling at the sequencing stage can give rise to similar copy number histograms with power law behaviour.

As copy numbers are positive integers, we modeled the distribution of copy numbers z with a discrete broken power law. The discrete probability distribution with the mass function following a power law is called zeta distribution. For the broken power law, we adopted the term *broken zeta distribution*, which we define as having the following probability

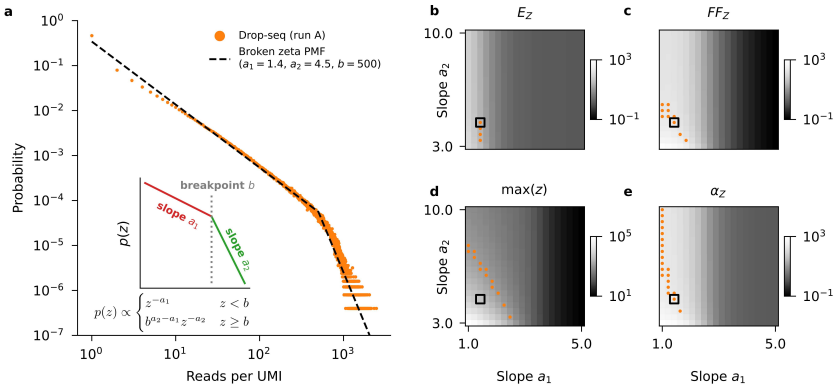

Figure 13: Broken zeta model can fit observed amplification distribution. a: Observed amplification distribution for Drop-seq (same as in Figure 12c) (orange dots) and the PMF of a broken zeta model (black line). The inset illustrates the parameters of the broken zeta model. **b:** The heatmap shows how the two slope parameters affect the mean of the broken zeta distribution. For each combination of a_1 and a_2, we sampled from a broken zeta model with these slope parameters and fixed $b = 500$. We used the same sample size as in the observed data in (a). The orange dots highlight the simulations yielding the sample mean close the observed mean ($\pm 10\%$). The black square shows the simulation corresponding to the fit shown in (a). **c–e:** Same as (b), but showing the sample Fano factor, sample maximum, and sample α_Z.

mass function (PMF):

$$p(z) \propto \begin{cases} z^{-a_1} & z < b \\ b^{a_2-a_1} z^{-a_2} & z \geq b \end{cases}, \tag{45}$$

where $a_1 > 0$ and $a_2 > 0$ are negative slopes of the PMF in log-log coordinates, and $b \in \mathbb{N}$ is the breakpoint between the two slopes. (Figure 13a, inset). We could choose the values for these three parameters such that the broken zeta distribution approximately matched the observed copy number histograms. For example, we obtained a good match for the Drop-seq protocol using $a_1 = 1.4$, $a_2 = 4.5$, and $b = 500$ (Figure 13a).

The fitted model could reproduce several key statistics of the experimental data, such as the mean, the variance, and the Fano factor (Figure 13b–e). However, the broken zeta distribution produced sample maxima that were larger than empirically observed maxima (given the same sample size) (Figure 13d). This is a limitation of the broken zeta model as it tends to allocate non-zero probability mass to very high copy numbers that are not observed in practice. A more flexible model that limits the probability of very large copy numbers could potentially fit the data even better, but we considered the broken zeta distribution sufficient for our purposes.

4.2.7 Compound NB model with broken zeta amplification captures trends in read count data

The compound NB model (Equations 35–38) together with the broken zeta amplification distribution (Equation 45) provides a generative probabilistic model of the read counts in a biologically homogeneous population. To confirm that the model gives rise to realistic data, we used it to sample read counts and compared them to observed read count histograms in a biologically homogeneous dataset (Figure 14). We used the same dataset as above in Figure 10. For the amplification distribution, we used broken zeta parameter settings ($a_1 = 0.36$, $a_2 = 5.1$, $b = 56$)

that led to an amplification model with $\alpha_Z = 50$ and $\mathbb{E}[Z] = 30$, as we showed earlier that these values fit the protein-coding genes in this dataset well (Figure 10).

We found that the empirical count distributions of real genes (Figure 14a–f, grey) could be well matched by the compound NB model (Figure 14a–f, orange). Note that there is only one free parameter per gene in our simulation: p_g, the fraction of RNA molecules taken up by this gene. The entire mass function is then determined by this single parameter. While the compound model did not fit every example gene perfectly (Figure 14b,e), it correctly captured the shapes of the distributions. In particular, for low-expression genes, the compound model predicted strong zero inflation and monotonically decreasing probability of non-zero counts (Figure 14a–b), while predicting a bell-shaped distribution without excess zeros for high expression genes (Figure 14f).

In order to study these patterns more systematically, we fitted a zero-inflated negative binomial (ZINB) distribution to the count histograms of each gene separately. ZINB models have been used to model read counts before (Lopez et al., 2018), as read count data commonly exhibit zero-inflation compared to NB (Cao et al., 2021; Chen et al., 2018). A ZINB distribution has three parameters: the mean μ, the inverse overdispersion θ, and the zero-inflation parameter ψ. Its mass function is simply a negative binomial mass function with additional mass ψ on zero:

$$p(z) = \begin{cases} \psi + (1-\psi) \cdot p_{\mathrm{NB}}(0, \mu, \theta) & \text{for } z = 0 \\ (1-\psi) \cdot p_{\mathrm{NB}}(z, \mu, \theta) & \text{for } z > 0 \end{cases} \tag{46}$$

where p_{NB} is the NB probability mass function (see Methods). The ZINB distribution reduces to the NB distribution when $\psi = 0$. As in NB, the overdispersion parameter θ controls the shape of the distribution: $\theta = 1$ corresponds to the geometric distribution with monotonically decreasing $p(x)$, while higher values of θ result in more Poisson-like bell shapes ($\theta = \infty$ corresponds to the Poisson case).

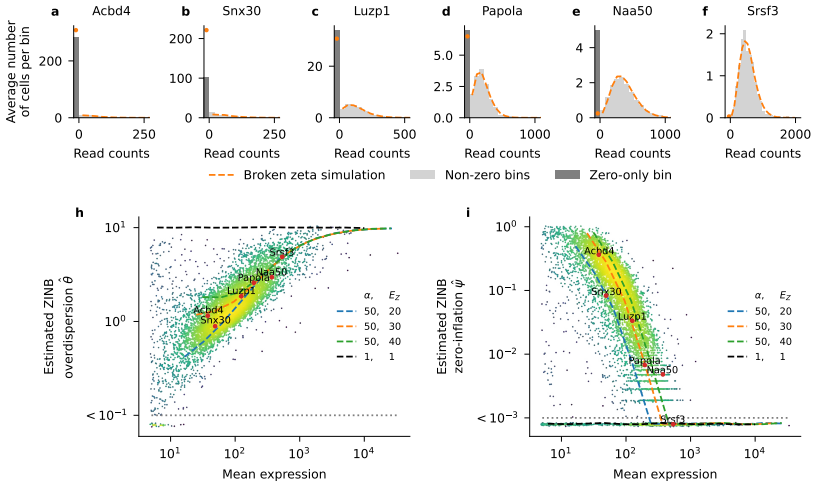

Figure 14: Broken zeta compound model simulations reproduce trends in read counts. Using a homogeneous subset ($n = 1\,049$) of the mouse cortex data (from Figure 10), we fitted a zero-inflated negative binomial (ZINB) distribution to each gene individually. We also used the broken zeta model ($a_1 = 0.36$, $a_2 = 5.1$, $b = 56$ to sample read counts with a given mean expression. **a–f:** Each panel shows the observed read counts for a certain gene (gray histogram) and the histogram of counts sampled from the broken zeta model (orange line). Exact zeros are shown in a separate bin (orange dots, dark-gray bar). To show the zero-bin (width 1) and non-zero bins (width $\gg 1$) on the same scale, the y-axis shows the average count per bin. Genes are ordered from left to right by mean expression. Note that with higher expression, fraction of zeros decreases, and the histogram shape changes from a geometric-looking distribution to a Poisson-looking distribution. **h:** Estimated ZINB overdispersion parameter $\hat{\theta}$ as a function of mean expression. Each dot is a gene, colored by local density of points. Values $\hat{\theta} < 10^{-1}$ were clipped. Only genes with mean expression ≥ 5 are shown. Colored lines show $\hat{\theta}$ for samples from four different broken zeta models. For each model, we sampled counts over a range of expression fractions p_g for 10^5 cells, each with fixed sequencing depth of $n_c = 100\,000$. We used overdispersion $\theta = 10$ for all genes. Broken zeta parameters: see Table 4. The black line corresponds to a negative binomial distribution (UMI model without amplification). Red dots highlight genes from panels (a)–(f). **i:** Estimated ZINB zero-inflation parameter $\hat{\psi}$ as a function of mean expression. Otherwise same as (h); values $\hat{\psi} < 10^{-3}$ were clipped.

By fitting the ZINB model, we obtained independent estimates $\hat{\theta}_g$ and $\hat{\psi}_g$ for each gene (Figure 14h–i). These estimates exhibited the same two patterns illustrated above for single genes. First, with increasing mean expression, genes tended to have a higher $\hat{\theta}_g$, corresponding to smaller variance, and transitioned from a geometric-like to a Poisson-like shape. Second, with increasing mean expression, genes tended to have a lower $\hat{\psi}_g$, corresponding to less pronounced zero inflation.

The ZINB model cannot explain these trends, as all parameters θ_g and ψ_g can be chosen independently. In contrast, our compound model naturally gives rise to both effects. To demonstrate this, we repeated the ZINB fitting procedure on counts sampled from various compound NB models (see Methods for the broken zeta parameters), and reproduced both observations over a wide range of mean expressions (Figure 14h–i, colored lines). As expected, the model matching this dataset's amplification parameters ($\alpha_Z = 50$ and $\mathbb{E}[Z] = 30$, orange line, cf. Figure 10) provided the best match to the bulk of the distribution.

As a sanity check, sampling read counts from a NB model without amplification ($\alpha_Z = 1$) and fitting ZINB distribution to the resulting samples recovered the original parameters (Figure 14h–i, black lines): constant overdispersion $\theta = 10$ and absent zero inflation $\psi = 0$. This again shows that a NB model without amplification cannot describe the properties of the read count data. However, our results suggest that it is not necessary to include explicit zero-inflation like in a ZINB model, as it is naturally arising through the compound model.

4.3 Chapter Discussion

In this paper, we derived a parsimonious and theoretically grounded statistical model describing scRNA-seq read count data without UMIs. Furthermore, we showed that our compound model leads to analytic compound Pearson residuals, a fast, simple, and effective normalization approach for non-UMI data.

Despite the popularity of UMI protocols (Svensson et al., 2020a),

full-length non-UMI protocols such as Smart-seq2 (Picelli et al., 2013) remain relevant as they have higher sensitivity (Ziegenhain et al., 2017; Ding et al., 2020) and allow quantification of reads over full transcripts. This makes read count data indispensable for detection of splicing variants (Feng et al., 2021) or profiling of complex tissues with rare cell types (Tasic et al., 2018; Yao et al., 2021). The recently developed Smart-seq3/Smart-seq3xpress protocols (Hagemann-Jensen et al., 2020, 2022) contain UMIs on the 5'-end reads but do not have UMIs on internal reads, so our treatment remains relevant for Smart-seq3/Smart-seq3xpress as well.

While UMI counts can be modeled by a Poisson or a negative binomial (NB) distribution (Grün et al., 2014; Chen et al., 2018; Hafemeister and Satija, 2019; Townes et al., 2019; Svensson, 2020; Grün, 2020; Sarkar and Stephens, 2021; Rosales-Alvarez et al., 2023; Neufeld et al., 2023), read counts can not (Chen et al., 2018; Cao et al., 2021). Instead, they are often modeled by a more flexible zero-inflated negative binomial distribution (ZINB) (Pierson and Yau, 2015; Zappia et al., 2017; Chen et al., 2018; Risso et al., 2018; Lopez et al., 2018). However, this leaves unexplained what causes zero inflation and why there are relationships between the gene-specific ZINB parameters, such as less zero inflation for higher mean expression (Figure 14i).

Our compound model answers these questions. We showed that read counts in biologically homogeneous data can be well described by a compound negative binomial distribution, arising from simple statistical assumptions about the amplification and sequencing processes. Furthermore, we showed empirically that the distribution of copy numbers approximately follows a broken zeta distribution. Together, our compound NB model with amplification modeled by broken zeta yields a generative model reproducing zero-inflation and overdispersion patterns similar to what is observed in read count data. Compared to the ZINB model with three per-gene parameters (Equation 46), our model contains only one free per-gene parameter (Equations 35–38), and the varying zero-

inflation and overdispersion naturally emerge as a function of a gene's mean expression.

We observed that the distribution of copy numbers in UMI-containing data followed a similar shape across various protocols (Figure 12), implying that this is a general property of scRNA-seq data. We argued that this shape could be described by a broken power law, and hence we modelled it with a broken zeta distribution. This model is phenomenological, but previous work on mechanistic modelling of PCR amplification followed by Poisson sampling showed that these processes can give rise to similar copy number distributions (Best et al., 2015). We note that fitting parameters for power-law-like data is intrinsically difficult: common approaches such as least squares often return unstable estimates due to low-probability events (Clauset et al., 2009), which is why we avoided automatic parameter fitting. Instead, we qualitatively showed that the broken zeta model can give rise to realistic read count distributions.

Our compound NB model did not describe all genes perfectly: we found that a subset of genes, mostly pseudogenes, did not follow the compound model, but rather behaved as if they were not amplified (Figures 10 and 11). Similar bimodal patterns in gene variance have been be observed in previous works (e.g. Brennecke et al., 2013; Ziegenhain et al., 2017). Pseudogenes are copies of functional genes that contain a mutation making the copy dysfunctional. We can only speculate about the reason causing pseudogene read counts to have less variance: they may behave differently during amplification or sequencing, or perhaps their counts are an artifact of the alignment algorithm (all datasets we analyzed used STAR (Dobin et al., 2013)). In practice, such pseudogenes have less variance than expected under the compound NB model, so will be filtered out by the gene selection step in our suggested workflow (Figure 11a).

On the practical side, we used the compound NB model to derive a fast and theory-based normalization procedure for read counts: compound Pearson residuals. They constitute an extension of the UMI Pear-

son residuals normalization, that has proven to be effective for gene selection and normalization of UMI data (Hafemeister and Satija, 2019; Lause et al., 2021). We showed that compound Pearson residuals work well for processing complex read count datasets, leading to a biologically meaningful gene selection and embeddings. Importantly, we also showed that normalization and gene selection using the non-compound UMI Pearson residuals leads to suboptimal results on read count data, underscoring the importance of an adequate statistical model.

Compound Pearson residuals only require to set the α_Z parameter of the amplification distribution. Whereas α_Z can be observed directly from the copy number distribution for UMI-containing data (i.e., reads per UMI, Table 3), it is unknown in a Smart-seq2 experiment. Reassuringly, we found that the results of compound Pearson residuals do not strongly depend on the exact value of α_Z — as long as it is set within a reasonable range $\gg 1$, such as $\alpha_Z \in [10, 1000]$. When working with Smart-seq2 data, we recommend using $\alpha_Z = 50$ by default. Furthermore, it is possible to empirically adjust α_Z to a given dataset from any sequencing protocol. Indeed, under the common assumption that most genes are not differentially expressed, the majority of genes should have residual variance close to one. Thus, adjusting α_Z until this condition is fulfilled will typically lead to a reasonable setting (Supplementary Figure S13).

Typical approaches to read count data normalization consist of scaling read counts by a size factor to account for sequencing depth (CPM: counts per million) and sometimes gene length (TPM: transcripts per million (Li and Dewey, 2011), or RPKM: reads per kilobase per million (Mortazavi et al., 2008)), followed by a log-transform (Luecken and Theis, 2019; Andrews et al., 2021; Slovin et al., 2021). Various methods have been suggested to estimate the required size factors, going beyond CPM/TPM/RPKM (Vallejos et al., 2017): via spike-ins (Brennecke et al., 2013; Lun et al., 2017), cell pooling (Lun et al., 2016), housekeeping genes (Andrews et al., 2021), separate scaling for groups

of genes (Bacher et al., 2017), or a Bayesian approach (Tang et al., 2020). However, all of these methods depend on the log-transform for variance stabilization, which is inherently limited (Lun, 2018) and fails to fully stabilize the variance (Ahlmann-Eltze and Huber, 2023). In contrast, our compound Pearson residuals use the mean-variance relationship that follows from simple statistical assumptions, and the resulting residuals are variance-stabilized by design and do not require any explicit normalization by the gene length.

Two existing methods aim to transform read counts so that their distribution matches the distribution of UMI counts. Census (Qiu et al., 2017) linearly scales the read counts within each cell to set the mode of the count distribution to 1, while qUMI (Townes and Irizarry, 2020) performs quantile normalization within each cell to transform the entire distribution to the typical shape of within-cell UMI counts. In both cases, the transformations are heuristics not based on any generative statistical model, and the transformed data still require UMI-specific normalization. In contrast, our compound Pearson residuals perform necessary normalization directly on the read counts. In practice, we observed that qUMI and Census lead to comparable normalization results as our compound Pearson residuals, but our method follows from an explicit statistical model that offers theoretical insights into the data generation process underlying read counts. For example, as described above, our model captures previously unexplained patterns in the zero-inflation and overdispersion of read count data.

One limitation of our model is that it assumes that the amplification distribution is the same for all genes and cells and uses the single amplification parameter α_Z shared by all cells. This is not the case in Census and qUMI, which both use cell-specific adjustments. Reassuringly, we did not observe strong cell-to-cell variability in α_Z estimates (Supplementary Figure S17), and furthermore found that α_Z correlated with total counts per cell (Supplementary Figure S18) — a factor which our model explicitly accounts for.

In summary, we show that the compound NB distribution is the appropriate statistical model for read count data, naturally giving rise to compound Pearson residuals as an effective, convenient and theoretically motivated way of data preprocessing.

4.4 Chapter Methods

Code and data availability The datasets generated and/or analysed during the current study are publicly available as described below in Methods (Section 4.4.1). All analysis code is available under the GNU General Public License v3.0 at `https://github.com/berenslab/read-normalization`, and is archived in the Zenodo repository `https://zenodo.org/doi/10.5281/zenodo.12806891`.

4.4.1 Datasets and preprocessing

Our example read count dataset throughout this paper is the mouse brain dataset from Tasic et al. (2018), GEO accession GSE115746. It contains cells from the primary visual cortex (VISp) and the anterior lateral motor area (ALM), and was sequenced with Smart-seq2. We downloaded the data for both areas from `https://portal.brain-map.org/atlases-and-data/rnaseq/mouse-v1-and-alm-smart-seq` and used only the exonic counts and applied the same cell filtering as Tasic et al. (2018), leading to 23 822 cells and 42 776 genes with at least one count. We used the Python package `mygene` to query the `mygene.info` database (Wu et al., 2013) for gene type annotations (`type_of_gene` field) with the Entrez gene identifiers. We queried `BioMart` to obtain transcript lengths for all Ensemble mouse genes (database *GRCm39*, column 'Transcript length (including UTRs and CDS)', Cunningham et al. (2022)).

From these data, we assembled a biologically homogeneous subset by selecting only cells from one of the largest neuronal clusters (named *VISp Penk Col27a1* by Tasic et al. (2018), 1 049 cells, 33 914 detected

genes). These data were used without further filtering in Figures 10 and 14 and Supplementary Figures S8 and S9.

To assemble a heterogeneous dataset, we took the full dataset but filtered out genes that were detected in less than 5 cells as in previous work on UMI Pearson residuals (Hafemeister and Satija, 2019; Lause et al., 2021), leading to 23 822 cells and 38 510 genes.

To study copy number distributions across protocols, we used the following UMI datasets:

- Mouse embryonic stem cells profiled by CEL-seq2, Drop-seq, MARS-seq, and SCRB-seq (Ziegenhain et al., 2017); GEO accession GSE75790;
- Mouse fibroblasts profiled with Smart-seq3 paired-end (Johnsson et al., 2022); accession E-MTAB-10148, sample `plate2`;
- Mouse fibroblasts profiled with Smart-seq3 single-end (Hagemann-Jensen et al., 2020); accession E-MTAB-8735, sample `Smartseq3.Fibroblasts.smFISH`;
- HEK293 cells profiled with Smart-seq3Xpress (Hagemann-Jensen et al., 2022); accession E-MTAB-11467.

For UMI deduplication, we used Hamming distance correction with a threshold of 1. See Table 3 for numbers of cells and UMIs per dataset. The reads-per-UMI tables are available at `https://zenodo.org/record/8172702`.

We used `scanpy 1.9.0` (Wolf et al., 2018) and `anndata 0.8.0` (Virshup et al., 2021) for all scRNA data handling in `Python 3.8.10`, along with `sklearn 1.0.2` (Pedregosa et al., 2011), `numpy 1.21.5` (Harris et al., 2020), and `matplotlib 3.5.1` (Hunter, 2007).

4.4.2 Simulation study

To generate a realistic validation dataset with ground truth marker genes *(Simulation I)*, we simulated read counts based on the Tasic et al. (2018) read counts X_{cg} as follows. For each gene g, we computed the average

expression fractions $p_g = \sum_c X_{cg} / \sum_{cg} X_{cg}$. For a set G of 14 well-known brain cell type marker genes (Tasic et al., 2016) and each of the 133 clusters in the Tasic et al. (2018) data, we computed the within-cluster fraction p_{ig}, where i is the cluster index. For each cell c, we computed its total read counts $n_c = \sum_g X_{cg}$ and assumed total UMI counts per cell $n_c^{\mathrm{UMI}} = n_c/100$. We then generated UMI counts as $k_{cg} \sim \mathrm{NB}(\mu_{cg}^{\mathrm{UMI}}, \theta = 100)$, where

$$\mu_{cg}^{\mathrm{UMI}} = \begin{cases} n_c^{\mathrm{UMI}} \cdot p_{i(c)g} & \text{for } g \in G \\ n_c^{\mathrm{UMI}} \cdot p_g & \text{for } g \notin G \end{cases}, \tag{47}$$

where $i(c)$ denotes cluster assignments of cell c. In words, only the marker genes from G were allowed to differ between clusters. We then simulated the amplification of each UMI by drawing copy numbers from the shifted geometric distribution $z_i \sim \mathrm{Z} = \mathrm{Geom}_+(\mu = 100)$, which corresponds to amplification with $\mathbb{E}[Z] = 100$ and $\alpha_Z = 199$ (see below). We finally summed the copy numbers for each gene and cell to obtain read counts $X_{cg} = \sum_{i=1}^{k_{cg}} z_i$ (Equation 35). After filtering out genes with less than 5 cells as above, *Simulation I* yielded 23 822 cells and 30 652 genes.

To obtain a second validation dataset with a richer cluster structure and ground truth cell types *(Simulation II)*, we used the same simulation setup as above, but allowed all genes to have cluster-specific fractions, i.e.

$$\mu_{cg}^{\mathrm{UMI}} = n_c^{\mathrm{UMI}} \cdot p_{i(c)g}. \tag{48}$$

After filtering as above, *Simulation II* yielded 23 822 cells and 30 576 genes.

Both simulations generated copy numbers z_i from the shifted geometric distribution $Z = \mathrm{Geom}_+(\mu = 100)$, which is equivalent to $Z = \mathrm{NB}(\mu = 99, \theta = 1) + 1$, with $z_i \in \mathbb{N}_+$ being positive integers. The variance of the negative binomial is equal to $99 + 99^2/1 = 9900$ and

$\mathbb{E}[Z] = 100$, so the Fano factor is 99, leading to $\alpha_Z = \mathbb{E}[Z] + \mathbb{FF}[Z] = 199$.

4.4.3 Mathematical details of the compound negative binomial model

We use the term *compound Poisson/NB distribution* to describe a discrete random variable that is constructed as a sum over a random number of i.i.d. terms. A compound model has an 'inner' and an 'outer' distribution: The inner distribution generates the i.i.d. summation terms (Equation 36), while the outer distribution governs the number of terms to be summed (Equation 37). This setup is known under various names: Johnson et al. (2005) uses the term *stopped-sum* distribution. When the outer distribution is the Poisson distribution, the compound model is known as compound Poisson (Adelson, 1966), stuttering Poisson (Kemp, 1967; Moothathu and Kumar, 1995), or generalized Poisson (Feller, 1943) distribution.

Note that the term *compound distribution* can also have a different meaning: for example, in their work on qUMI normalization, Townes and Irizarry (2020) used the term 'compound Poisson model' to describe a Poisson model with rate parameter λ governed by another distribution.

The expectation of a compound random variable $X = \sum_{i=1}^{k} z_i$ with inner distribution $z_i \sim Z$ and outer distribution $k \sim K$ can be obtained as follows:

$$
\begin{aligned}
E[X] &= \mathbb{E}_K\big[\mathbb{E}_X[X \mid K]\big] && (49)\\
&= \mathbb{E}_K\big[\mathbb{E}_X[z_1 + z_2 + \cdots + z_k \mid K]\big] && (50)\\
&= \mathbb{E}_K\big[k \cdot \mathbb{E}_X[Z]\big] && (51)\\
&= \mathbb{E}[K] \cdot \mathbb{E}[Z]. && (52)
\end{aligned}
$$

The variance can be computed similarly:

$$\begin{aligned}
\mathrm{Var}[X] &= \mathbb{E}_K\big[\,\mathrm{Var}[X \mid K]\big] + \mathrm{Var}_K\big[\mathbb{E}[X \mid K]\big] && (53)\\
&= \mathbb{E}_K\big[\,\mathrm{Var}[z_1 + z_2 + \cdots + z_k \mid K]\big] && (54)\\
&+ \mathrm{Var}_K\big[\mathbb{E}[z_1 + z_2 + \cdots + z_k \mid K]\big] && (55)\\
&= \mathbb{E}_K\big[k \cdot \mathrm{Var}[Z]\big] + \mathrm{Var}_K\big[k \cdot \mathbb{E}[Z]\big] && (56)\\
&= \mathbb{E}[K] \cdot \mathrm{Var}[Z] + \mathrm{Var}[K] \cdot \mathbb{E}[Z]^2. && (57)
\end{aligned}$$

Together, this leads to the following mean-variance relationship:

$$\mathrm{Var}[X] = \mathbb{E}[X] \cdot \frac{\mathrm{Var}[Z]}{\mathbb{E}[Z]} + \frac{\mathrm{Var}[K]}{\mathbb{E}[K]^2} \cdot \mathbb{E}[X]^2. \tag{58}$$

We use the negative binomial (NB) distribution as outer distribution in our compound model. The probability mass function for the NB distribution can be parametrized in several different ways. We use

$$p_{\mathrm{NB}}(k, \mu, \theta) = \frac{\Gamma(k+\theta)}{k!\,\Gamma(\theta)} \Big(\frac{\mu}{\mu+\theta}\Big)^k \Big(\frac{\theta}{\theta+\mu}\Big)^\theta, \tag{59}$$

where μ is the mean and θ is the overdispersion parameter. The variance is then given by $\mathrm{Var}[K] = \mathbb{E}[K] + \mathbb{E}[K]^2/\theta = \mu + \mu^2/\theta$.

Plugging the mean-variance relationship of K into the mean-variance relationship of X, we finally get

$$\begin{aligned}
\mathrm{Var}[X] &= \mathbb{E}[X] \cdot \frac{\mathrm{Var}[Z]}{\mathbb{E}[Z]} + \frac{\mathbb{E}[K] + \mathbb{E}[K]^2/\theta}{\mathbb{E}[K]^2} \cdot \mathbb{E}[X]^2 && (60)\\
&= \mathbb{E}[X] \cdot \frac{\mathrm{Var}[Z]}{\mathbb{E}[Z]} + \frac{\mathbb{E}[X]^2}{\mathbb{E}[K]} + \frac{\mathbb{E}[X]^2}{\theta} && (61)\\
&= \mathbb{E}[X] \cdot \underbrace{\left(\frac{\mathrm{Var}[Z]}{\mathbb{E}[Z]} + \mathbb{E}[Z]\right)}_{\alpha_Z} + \frac{E[X]^2}{\theta}, && (62)
\end{aligned}$$

where we used the fact that $\mathbb{E}[X] = \mathbb{E}[Z] \cdot \mathbb{E}[K]$. The relationship in Equation 62 yields the lines shown in Figure 10a.

From here we can obtain the relationship between the mean of X and the Fano factor of X:

$$\mathbb{FF}[X] = \frac{\mathrm{Var}[X]}{\mathbb{E}[X]} = \alpha_Z + \frac{\mathbb{E}[X]}{\theta}, \tag{63}$$

shown as lines in Figure 10b. For $E[X] \ll \theta$, this reduces to $\mathbb{FF}[X] \approx \alpha_Z$.

To derive the relationship between the mean of X and the fraction of zero counts, we note that the inner distribution in our model is strictly positive ($z \geq 1$). Any zero count $X = 0$ must thus originate from a $k = 0$ from the outer NB distribution. As a result, we can derive the fraction of zero counts from the NB probability mass function (Equation 59):

$$P(X = 0) = p_{\mathrm{NB}}(K = 0) = \left(\frac{\theta}{\theta + \mathbb{E}[K]}\right)^{\theta} = \left(\frac{\theta}{\theta + \mathbb{E}[X]/\mathbb{E}[Z]}\right)^{\theta}. \tag{64}$$

This relationship is shown as lines in Figure 10c. For $\theta \to \infty$, this converges to the Poisson case $P(X = 0) = e^{-\mathbb{E}[K]} = e^{-\mathbb{E}[X]/\mathbb{E}[Z]}$.

4.4.4 Compound Pearson residuals

For gene selection with compound Pearson residuals, we computed $\hat{\mathbb{E}}[X_{cg}]$ from the filtered read count matrix X (Equation 40) and then obtained residuals using Equation 43. We selected 3 000 highly variable genes (HVGs) with the highest residual variance. To normalize, we then subset the raw count data matrix to the HVGs, and computed compound Pearson residuals again on that subset, and used these re-computed residuals for further analysis (consistent with our previous work, Lause et al. (2021)). Using the residuals computed from the full data matrix and subsetting them to HVGs led to very similar results.

Unless otherwise stated, we used $\alpha_Z = 50$ and $\theta = 100$ for computing residuals, and clipped residuals to $\sqrt{n}$ where n is the number of cells, following Hafemeister and Satija (2019) (see Lause et al. (2021) for a motivation for this heuristic).

4.4.5 Census counts and qUMIs

We obtained both Census counts and qUMIs via their official `R` implementations using `R 4.1.3`. To obtain Census counts, we used `bioconductor-monocle 2.22.0` (Huber et al., 2015; Qiu et al., 2017). To obtain qUMIs, we used `quminorm 0.1.0` from `http://github.com/willtownes/quminorm/` (Townes and Irizarry, 2020). As both methods expect TPMs as input, we subset the (Tasic et al., 2018) data to the 27 841 genes for which length annotations were available (see above), and computed TPM from read counts X_{cg} and gene lengths l_g (in kilobase) as

$$\mathrm{TPM}_{cg} = \frac{X_{cg}/l_g}{\sum_g X_{cg}/l_g} \cdot 1\,000\,000 \tag{65}$$

Running Census on the full matrix was very slow (>24 h), so we split the TPM matrix into batches of 1000 cells. This substantially sped up the computation. Filtering Census counts and qUMIs for genes with at least 5 cells yielded 23 822 cells and 25 248 genes.

4.4.6 t-SNE visualizations

As basis for all t-SNE embeddings, we computed the first 1 000 principal components (PCs) of the HVGs residuals with `sklearn 1.0.2` (Pedregosa et al., 2011). For all t-SNE embeddings, we used `openTSNE 0.6.0` (Poličar et al., 2019) with default settings unless otherwise stated. To ensure comparability between the t-SNE embeddings in Supplementary Figure S13, we used the first two PCs of the HVG residuals computed with $\alpha_Z = 10$ (panel C) as shared initialization after scaling them with `openTSNE.initialization.rescale()`.

To visualize expression strength of a given gene across the t-SNE map(Figure 11c), we used square-root-transformed depth-normalized counts

$$S_{cg} = \sqrt{m \cdot \frac{X_{cg}}{\sum_i X_{ci}}}, \tag{66}$$

where m is the median row sum of X.

4.4.7 Fitting zero-inflated negative binomial (ZINB) models to single genes

To obtain per-gene estimates of overdispersion $\hat{\theta}_g$ and zero inflation $\hat{\psi}_g$ in the absence of biological variability, we fitted a ZINB model to the raw read counts of each gene in the *VISp Penk Col27a1* cluster ($n = 1\,049$ cells). We used the `ZeroInflatedNegativeBinomialP.fit_regularized()` function from `statsmodels 0.13.2` (Seabold and Perktold, 2010) with default parameters. Only the 11 549 genes with within-cluster mean expression >5 were included because low-expression genes suffered from unstable parameter estimates. All genes with fitting warnings ($n = 2\,064$), fitting errors ($n = 22$) or invalid resulting estimates $\hat{\psi}_g > 1$ ($n = 2\,359$) were excluded, such that 7 104 genes with valid converged estimates remained for further analysis and are shown in Figures 14h–i.

We applied the same fitting procedure to the simulated read counts shown as lines in Figure 14 (see below for simulation details). Here, 62 out of 100 simulated genes had a mean expression >5 and were used for fitting. 5 of them resulted in invalid $\hat{\psi}_g > 1$ values and were excluded, such that 57 simulated genes remained for plotting and analysis.

4.4.8 Sampling copy numbers from the broken zeta model to simulate compound model read counts

The broken zeta model we describe in Equation 45 is the discrete version of the broken power law. A continuous probability distribution with probability density following a power law is called the Pareto distribution. Its discrete analogue is known under various names including Riemann zeta distribution (or simply zeta distribution), discrete Pareto distribution, and Zipf distribution(Johnson et al., 2005). We therefore refer to the discrete broken power law distribution as *broken zeta distribution*. While continuous broken power law distributions are commonly

$\mathbb{E}[Z]$		α_Z				
target	observed	target	observed	a_1	a_2	b
1	1.0	1	1.0	–	–	–
20	20.1	50	50.4	0.96	15.1	91
30	30.2	50	51.4	0.36	5.1	56
40	37.9	50	50.5	0.01	17.6	71

Table 4: Broken zeta parameters used for compound model read count simulations. Each row in the table corresponds to one of the four compound model simulations shown in Figure 14. Observed values refer to the obtained sample mean and sample variance ($n \approx 2$ billion, see text for simulation details). The model with $\alpha_Z = 1$ corresponds to the UMI case with constant copy number $z_i = 1$, and thus has no broken zeta parameters.

used in astrophysics (Jóhannesson et al., 2006), we are not aware of any prior use of discrete broken power law distributions for statistical modeling.

For the simulations in Figures 13 and 14, we sampled copy numbers from a given broken zeta distribution. For that, we computed the approximate PMF for a limited support $z \in \{1, 2, \ldots, 10^5\}$ with Equation 45, and normalized the resulting probabilities to sum to 1. This way we did not need to compute the normalization constant in Equation 45.

For the simulations of the Drop-seq copy number distribution in Figure 13b–e we used $n = 2\,506\,244$ samples per parameter set, which is the number of UMIs in the Drop-seq A dataset. We extended the support until 10^6 for this particular simulation, because we observed $\max(z_i) \approx 10^5$ for some of the more extreme parameter combinations.

In order to sample realistic read counts from four compound models with different combinations of mean copy number $\mathbb{E}[Z]$ and α_Z (Figure 14), we used a grid search over the broken zeta parameters a_1, a_2, and b to find parameter combinations that best matched the required values. Table 4 lists the parameters and amplification statistics of the chosen models. Across all parameter sets shown in Figure 14, we observed $\max(z_i) = 7\,179$ which was far below the end of the support.

We first sampled unique sequenced molecules k_{cg} from a negative bi-

nomial with $k_{cg} \sim \mathrm{NB}(p_g n_c, \theta)$ for 25 genes over a log-spaced range of expression fractions $p_g \in [10^{-8}, 10^{-1}]$. We did this for 10^5 cells, each with fixed sequencing depth of $n_c = 10^5$. We used overdispersion parameter $\theta = 10$ for all genes. This led to a total of $\sum k_{cg} = 2\,047\,196\,087$ simulated UMIs. Then, for each of them, we sampled a copy number z_i from the broken zeta model as described above, and summed over copy numbers for the same cell and gene to obtain a read count $X_{cg} = \sum_{i=1}^{k_{cg}} z_i$. We used the same set of simulated UMIs for the four compound model simulations shown in Figure 14 and Table 4.

Chapter Acknowledgements

We would like to thank Rickard Sandberg for discussions.

5 The art of seeing the elephant in the room

Publication note. This chapter is published in *PLOS Computational Biology* (Lause et al., 2024) and is available at `https://doi.org/10.1371/journal.pcbi.1012403` under a `CC-BY 4.0` license (`https://creativecommons.org/licenses/by/4.0/`).

Abstract

A recent paper in *PLOS Computational Biology* (Chari and Pachter, 2023) claimed that t-SNE and UMAP embeddings of single-cell datasets fail to capture true biological structure. The authors argued that such embeddings are as arbitrary and as misleading as forcing the data into an elephant shape. Here we show that this conclusion was based on inadequate and limited metrics of embedding quality. More appropriate metrics quantifying neighborhood and class preservation reveal the elephant in the room: while t-SNE and UMAP embeddings of single-cell data do not preserve high-dimensional distances, they can nevertheless provide biologically relevant information.

5.1 2D embeddings of single-cell data do make sense

In single-cell genomics, researchers often visualize data with 2D embedding methods such as t-SNE (van der Maaten and Hinton, 2008; Kobak and Berens, 2019) and UMAP (McInnes et al., 2018; Becht et al., 2019). Chari and Pachter (2023) criticize this practice: They claim that the resulting 2D embeddings fail to faithfully represent the original high-dimensional space, and that instead of meaningful structure these embeddings exhibit "arbitrary" and "specious" shapes. While we agree that 2D embeddings necessarily distort high-dimensional distances between data points (Nonato and Aupetit, 2018; Wang et al., 2023b), we believe that UMAP and t-SNE embeddings can nevertheless provide useful information. Here, we demonstrate that UMAP and t-SNE preserve cell neighborhoods and cell types, and that the conclusions of Chari and Pachter (2023) are based on inadequate metrics of embedding quality.

To illustrate their point that *t*-SNE and UMAP embeddings are arbitrary, Chari and Pachter (2023) designed Picasso, an autoencoder method that transforms data into an arbitrary predefined 2D shape, e.g., that of an elephant. The authors then compared four kinds of embeddings: the purposefully arbitrary elephant embedding, 2D PCA, *t*-SNE, and UMAP (Figure 15). For this, they used two metrics of embedding quality, both requiring class annotations: *inter-class correlation* measuring how well high-dimensional distances between class centroids are preserved in the 2D embedding and *intra-class correlation* measuring how well class variances are preserved. They found that across three scRNA-seq datasets, 2D PCA performed the best on those metrics, while the elephant embedding scored similar to or better than UMAP and *t*-SNE. We reproduced and confirmed these results (Figure 16a–b).

According to the authors, this means that *t*-SNE and UMAP are as arbitrary and as misleading as the Picasso elephant. Most online discussions and debates about their paper, including posts by the authors themselves, have prominently featured this argument and the powerful elephant metaphor to argue that "it's time to stop making *t*-SNE & UMAP" plots (Pachter, 2021). In this Comment, we focus exclusively

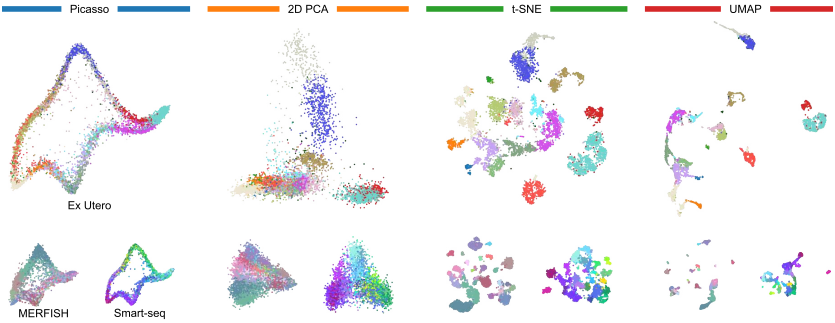

Figure 15: Evaluated embeddings. Large panels: `Ex Utero` dataset. Small panels: `MERFISH` and `Smart-seq` datasets. Colors correspond to cell types and are taken from Chari and Pachter (2023). See Chapter Methods for details.

on this argument and do not discuss the rest of the Chari and Pachter (2023) paper.

We believe that this argument is faulty because the metrics used by Chari and Pachter (2023) are insufficient and only quantify a single aspect: both metrics focus on preservation of *distances*, where 2D PCA was unsurprisingly the best. But there is more to embeddings than distance preservation. It is visually apparent in the resulting embeddings that t-SNE and UMAP separate cell types, while 2D PCA and Picasso elephant lead to strongly overlapping types (Figure 15), but neither of the two metrics quantified that. Biologists are often interested in cell clusters, and so preservation of cell neighborhoods and visual separation of meaningful cell groups are important properties of 2D embeddings.

To quantify these aspects neglected by Chari and Pachter (2023), we used four additional metrics, commonly employed in benchmark studies (Espadoto et al., 2019; Huang et al., 2022a; Wang et al., 2023a): k-nearest-neighbor (kNN) accuracy, kNN recall (Lee and Verleysen, 2009), the silhouette coefficient (Rousseeuw, 1987), and the adjusted mutual information (AMI) between clusters and class labels (Vinh et al., 2009).

The kNN accuracy quantifies how often the 2D neighbors are from the same class, while the kNN recall quantifies how often the 2D neighbors are the same as the high-dimensional neighbors. In both metrics, UMAP and t-SNE consistently and strongly outperformed PCA and Picasso elephant embeddings (Figure 16c–d, >90% vs. <62% accuracy, >15% vs. <5% recall for all datasets). Even though the kNN recall was below 40% for all methods (Figure 16d), kNN accuracy was always above 90% for both UMAP and t-SNE (Figure 16c). This means that even though UMAP and t-SNE are not able to preserve high-dimensional nearest neighbors exactly, the low-dimensional neighbors tend to be from a close vicinity in the high-dimensional space, have the same cell type, and hence allow reliable kNN classification. In contrast, 2D PCA and the Picasso elephant fail at that.

The silhouette coefficient and the AMI both evaluate to what extent

cell types appear as isolated islands in 2D. Specifically, the silhouette coefficient measures how compact and separated the given classes are in 2D, while the AMI evaluates how well clustering in 2D recovers the classes. In both metrics, t-SNE and UMAP strongly outperformed 2D PCA and Picasso elephant embeddings (Figure 16e–f, >0.3 difference in silhouette score, >0.25 difference in AMI), in agreement with the visual

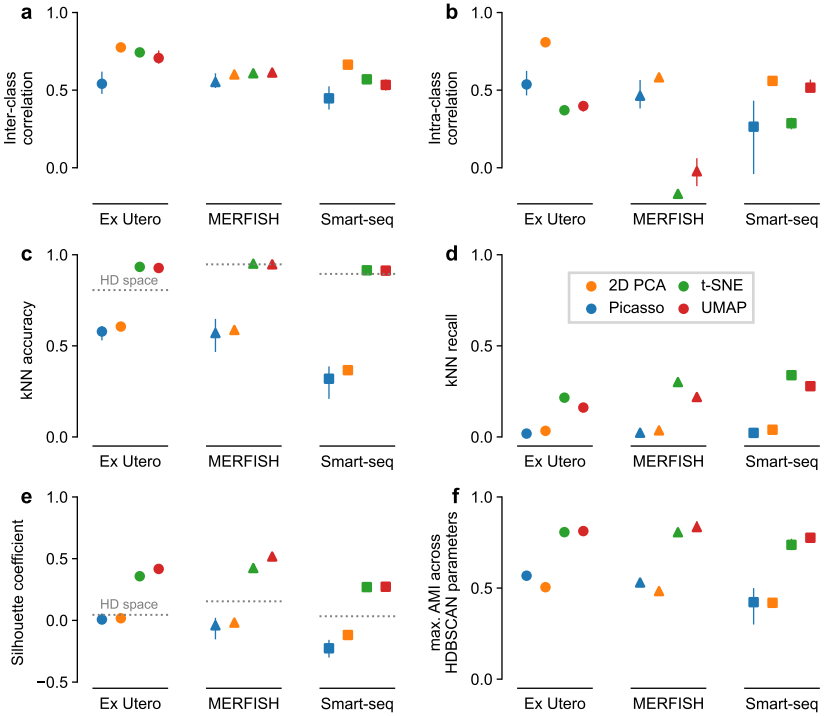

Figure 16: Embedding quality metrics. Panels correspond to metrics, colors correspond to embedding methods, marker shapes correspond to datasets. Averages over five runs, error bars go from the minimum to the maximum across runs. Dotted horizontal lines show the values of the metrics in the high-dimensional gene space. **a–b:** The two metrics from Chari and Pachter (2023), reproducing their Figure 7. **c–d:** kNN accuracy and kNN recall ($k = 10$). **e:** Silhouette coefficient. **f:** Maximum adjusted mutual information between classes and 2D clusters obtained with HDBSCAN using a range of hyperparameter values.

impression (Figure 15).

The kNN accuracy and the silhouette coefficient can also be computed directly in the high-dimensional gene space. We found that t-SNE and UMAP showed similar or higher kNN accuracy and much higher silhouette coefficient than the original high-dimensional space (Figure 16c,e). This suggests that high-dimensional distances suffer from the curse of dimensionality, and that it may in fact be undesirable to preserve them in 2D visualizations. Indeed, single-cell biologists rarely use multidimensional scaling (MDS), an embedding method explicitly designed to preserve distances, because MDS often fails to represent the cluster structure in the data. This further underscores why using only distance-preservation metrics, as Chari and Pachter (2023) did, is misguided.

All presented metrics except kNN recall rely on class labels, and our analysis, following Chari and Pachter (2023), used labels derived in original publications via clustering. Therefore, these labels do not necessarily correspond to biological ground truth, and could potentially lead to biased comparisons. To address this concern, we used negative binomial sampling based on the `Ex Utero` dataset to simulate a dataset with known ground truth classes. Analyzing this simulated dataset gave the same conclusions: 2D PCA scored the best in the distance-based correlation metrics of Chari and Pachter (2023), but only t-SNE and UMAP could separate the true classes, while Picasso and 2D PCA failed at that (Figure S19).

Taken together, our results point to the elephant in the room: Even though they are not designed to preserve pairwise distances, t-SNE and UMAP embeddings are not arbitrary and do preserve meaningful structure of single-cell data, especially local neighborhoods and cluster structure. Claiming that Picasso and t-SNE/UMAP are "quantitatively similar in terms of fidelity to the data in ambient dimension" (Chari and Pachter, 2023) is wrong. They are not.

That said, we do agree with Chari and Pachter (2023) that 2D visu-

alizations distort distances and should not be blindly trusted. Moreover, as Chari and Pachter (2023), we do not recommend to use 2D embeddings for any quantitative downstream analysis. However, paraphrasing George Box (Box, 1979), we can say that *all 2D embeddings of high-dimensional data are wrong, but some are useful*. Indeed, one can use 2D embeddings to form hypotheses about the data structure, ranging from data quality control and sanity-checking of any algorithmic output, to more general hypotheses about cluster separability, relationships between adjacent clusters, or presence of outlying clusters. Of course, any generated insight should then be validated in the high-dimensional data by other means. Here, our conclusion differs strongly from that of Chari and Pachter (2023): while they claim that UMAP and t-SNE are "counter-productive for exploratory [...] analyses", we endorse them for that very purpose.

5.2 Chapter Methods

Code and data availability Our code in Python is available at `https://github.com/berenslab/elephant-in-the-room`. All data is publicly available as described below.

Datasets and preprocessing We used the same datasets as Chari and Pachter (2023) (Table 5) and followed the same preprocessing steps. The `Ex Utero` data was already log-normalized. We filtered empty genes and cells, and selected the 2 000 most highly variable genes (HVGs) with `scanpy.pp.highly_variable_genes()` with default settings (Wolf et al., 2018). The `MERFISH` data was already normalized, and we only performed the `log1p()` transform. The `Smart-seq` data was already log-normalized and had HVGs selected, so we used it as is. Chari and Pachter (2023) additionally performed a standardization step on all datasets, which we omitted for simplicity, as it did not change the result qualitatively (see our Github repository for a direct comparison). Despite small differences in preprocessing choices, we obtained qualita-

tively very similar results in Figure 16a–b to what the original authors reported in their Figure 7.

Simulation We used negative binomial sampling to obtain a simulated version of the `Ex Utero` dataset with known ground-truth classes. For each cluster and each gene g in the original dataset, we computed the proportion p_g of UMI counts of this gene among all UMI counts in the cluster. For each cell c belonging to this cluster in the original data, we then sampled new counts $X_{cg} \sim \mathrm{NB}(\mu = n_c p_g, \theta = 10)$, where n_c is the cell's original total UMI count. Overdispersion parameter $\theta = 10$ leads to some additional variance compared to the Poisson distribution. This procedure preserved the number of genes, the number of cells, and all class abundances, and ensured realistic marginal distributions of simulated counts per cell and per gene. The counts of each simulated gene in each class followed an independent negative binomial distribution around the gene's mean expression in the original `Ex Utero` cluster. Finally, we performed the same preprocessing as above on the simulated counts (depth normalization, scaling normalized counts to 10 000 counts per cell, `log1p()` transform, `scanpy` default HVG selection).

Embeddings We used the high-dimensional gene space after preprocessing and gene selection as input to all embedding methods. For the elephant embeddings, we used the original Picasso code by Chari and Pachter (2023) with minimal adjustments needed to provide the random seed for reproducibility (`https://github.com/berenslab/picasso`). We ran Picasso for 500 epochs with default settings. For PCA, we used `scikit-learn 1.3.0` (Pedregosa et al., 2011) with default parameters. For t-SNE and UMAP, we followed Chari and Pachter (2023) and first reduced the preprocessed count matrices to 50 dimensions with PCA and used that as input to `openTSNE 1.0.1` (Poličar et al., 2024) and `umap-learn 0.5.5` with default parameters. The 50-dimensional PCA was used in no other part of the analysis. In all plots, we used the class labels and colors from Chari and Pachter (2023), except for minor ad-

Name	Cells	Genes	Classes	Source	Count matrix	Metadata
`Ex Utero`	6 205	19 588	19	Aguilera-Castrejon et al. (2021)	`normalized.assay85`	`MetaData.85`
`MERFISH`	6 963	254	25	Zhang et al. (2021)	10.22002/D1.2064	10.22002/D1.2063
`Smart-seq`	3 850	1 999	28	Kim et al. (2019)	10.22002/D1.2071	10.22002/D1.2067

Table 5: Datasets. `Ex Utero`: Files in GEO accession GSE149372. `MERFISH` and `Smart-seq`: DOIs.

justments to the `Ex Utero` colors, where we introduced four additional colors to make all classes discernible.

Embedding quality metrics Following Chari and Pachter (2023), we computed their intra- and inter-class correlation metrics using both L^1 and L^2 distances (see our Github repository for a direct comparison). As we did not observe qualitative differences between the two variants, we only showed L^2 results here, and also used L^2 distances for all other metrics.

For kNN accuracy, we used the k nearest neighbors in the 2D embedding to predict the class of each cell with a majority vote (this is essentially a leave-one-out cross-validation procedure). We reported raw accuracy here, but class-balanced accuracy gave qualitatively the same results (see our Github repository). For kNN recall, we computed (for each cell) the fraction of the k nearest neighbors in the 2D embedding that are also among the k nearest neighbors in the high-dimensional space. For both kNN metrics, we used $k = 10$, and averaged over all cells.

For the maximum AMI metric, we ran HDBSCAN (McInnes and Healy, 2017) from `scikit-learn` on each embedding for nine hyperparameter values `min_samples` = `min_size_clusters` $\in \{5, 10, 15, 20, 30, 40, 50, 75, 100\}$. All points that HDBSCAN left unclustered (noise points) we assigned to their nearest clusters. We then computed the adjusted mutual information (AMI) between each HDBSCAN result and the given cell type class labels, and picked the largest AMI. This way, the best performing hyperparameter was chosen for each embedding and each dataset.

The silhouette coefficient of each cell is defined as $(b - w)/\max(b, w)$ where w is the average distance to cells from the same class and b is the average distance to cells in the nearest other class. The silhouette coefficient is then averaged across all cells. We used `scikit-learn` to find kNNs, and to compute AMI and silhouette coefficients.

For all metrics that required a high-dimensional reference space for

comparison (inter-class and intra-class correlations, kNN recall), we used the same high-dimensional gene space that we used as input to the embedding methods.

Chapter Acknowledgements

We thank Erik van Nimwegen, Sebastian Damrich, and Pavlin Poličar for discussions.

6 Discussion

This thesis addressed two open challenges in neuroscience single-cell RNA-seq data: preprocessing and visualization. In the final Chapter, we will first review the strengths and limitations of the new preprocessing framework we propose, and how scRNA-seq preprocessing methods in general could profit from improved benchmarks. Then, we will summarize what we learned from the recent debates about 2D visualizations for scRNA-seq data, and make suggestions on how to ensure 2D embeddings do not mislead their users. Finally, we sketch out how neuroscience can build on top of the exciting scRNA-seq findings from the last decade — and what all of that means for the concept of cell types.

Strength and limitations of Pearson residual-based preprocessing. In Chapter 3 and 4 of this thesis, we presented two theory-aware methods to preprocess single-cell RNA sequencing data, both based on null model Pearson residuals. The concept is simple: We model the expected technical and statistical noise from the data generation process, and then fit this null model to observed count data. Our null model only accounts for non-biological factors, such that ideally, the residuals of this model will contain no more noise and all the biologically meaningful signal. That is why we propose Pearson residuals — with appropriate null models for UMI and non-UMI data — as a generic preprocessing framework for scRNA-seq data.

Even before we started to work on this method, a multitude of scRNA-seq preprocessing methods already existed. How does our contribution add value to the existing options? As detailed in Section 2.3.1, the most common preprocessing heuristic — global scaling — is simple, fast and seems to give acceptable results in many cases. However, several benchmarks showed that this performance is not optimal (Cole et al., 2019; Tian et al., 2019). This is likely due to the theoretical limitations and misspecified implicit assumptions of this method (Lun, 2018; Warton, 2018; Ahlmann-Eltze and Huber, 2023), leading to cases

in which global scaling does not achieve the desired normalization and variance stabilization, such that technical noise remains in the preprocessed data.

We showed that normalization by Pearson residuals avoids these problems both in theory and in practice. At the same time Pearson residuals are fast to compute and scale to datasets with millions of cells. Also, we made analytic Pearson residuals for UMI data available in `scanpy` (Wolf et al., 2018), the most popular `python` library for scRNA-seq processing. That way, Pearson residuals are an easily accessible and often superior alternative to the common global scaling normalization.

In contrast to other methods like *Sanity* (Breda et al., 2021) or *scTransform* (Hafemeister and Satija, 2019), Pearson residuals require fewer parameters to be fit and will usually be faster. As a result, Pearson residuals can only remove relatively simple noise structure from data: For example, our method does not allow the mean-variance relationship (for which the model eventually corrects) to change between genes, while both *scTransform* and *SCnorm* (Bacher et al., 2017) allow for that. This additional flexibility will lead these models to remove more variance from the data than our method. It is debatable if that is conceptually desirable, because a change in mean-variance relationship between genes has no obvious theoretical justification, and it can easily happen that biological variance or unexplained technical noise sources are inadvertently "explained away" by these flexible models. In fact, our experience with unexpected technical artifacts (Figures 7 and S6) has shown that simple models like ours can have the advantage that the data analyst will immediately notice if the assumptions of the model are violated, while other methods might just smooth over surprising structure. Also, in the case of our non-UMI count residuals, we saw that our model was able to predict and explain statistical patterns in the zero-inflation and overdispersion of non-UMI data that were unexplained before. This is another example of how our rather parsimonious model helped the understanding of scRNA-seq data in general.

However, the simple model of our normalization framework comes with limitations. Some well-known noise sources, like batch effects (Tung et al., 2017), are not part of the model. This might limit the applicability of Pearson residuals normalization in its current form for data from large, multi-sample experiments (although we were successful in using UMI residuals on such a dataset (Cao et al., 2021)). However, related work by Hafemeister and Satija (2019) showed that it might be sufficient to simply add a batch covariate to the null model.

As another drawback of our work, we only ran relatively small-scale benchmarks to validate our methods: For both UMI and non-UMI residuals, we provided proof-of-concept results on a heterogeneous, real-world dataset, and validation on more artificial data with known ground truth. However, while we compared to the most common alternative preprocessing pipelines, we could not cover all possible options. Also our evaluations confirmed that variance stabilization worked well, and the Pearson residuals representation allowed to separate cell populations of interest — but we did not evaluate on more involved downstream analysis task, like clustering or differential expression analysis.

The scRNA-seq field lacks task-driven benchmarks of preprocessing methods. This limitation points to the fact that comprehensive benchmarks that shed light on which preprocessing method performs best under certain circumstances remain an open gap in the field. Two relatively recent benchmarks (Booeshaghi et al., 2022; Ahlmann-Eltze and Huber, 2023) attempted to fill this gap with very different approaches: Booeshaghi et al. (2022) defined three desiderata for preprocessing (variance stabilization, sequencing depth normalization, monotonicity of the transform) and created a metric for each. They then evaluated how well each of ten preprocessing methods fulfilled these desiderata on > 500 scRNA-seq datasets. While the large number of datasets and the principled approach to start from desiderata is of value, the study had severe limitations: nine out of ten methods were global scaling methods, such that the diversity of available model-based nor-

malization methods was not well represented here. Also, a benchmark of desiderata metrics alone does not answer the question if and how strong the reported differences between methods affect downstream performance. Therefore, both the selection of normalization methods and the choice of metrics was too one-sided to draw final conclusions on which normalization methods are best.

In contrast, the benchmark by Ahlmann-Eltze and Huber (2023) selected methods from all major categories of normalization methods, including global scaling, null model residuals and Bayesian models. Also, their metrics focused on how the preprocessing affected the kNN graph of the data — a view of the data that is often used in downstream analysis. In that respect, they addressed the problems of the Booeshaghi et al. (2022) benchmark. Additionally, their metrics were unsupervised in the sense that they did not require manually annotated cell type labels. However, these metrics seemed to be not very sensitive, as only one of their metrics showed reliable differences between methods. This metric measured if the kNN graph was "consistent" across subsets of genes, i.e., whether the graph showed the same neighborhood structure irrespective of the gene used to compute it. However, from this metric it is hard to tell if a method is "consistently good" or "consistently bad": For example, if a method does not normalize for sequencing depth properly, its kNN graph might very consistently put cells with similar sequencing depth next to each other—but that would not be a desired outcome after normalization for exactly that factor.

In summary, both benchmarks fell short of a conclusive survey of scRNA-seq normalization method. Such a benchmark is much needed to inform the choices of practitioners that have to deal with noisy scRNA-seq data, potentially without being trained as statisticians or computer scientists. In order to fill this gap, future benchmarks should (i) include a representative set from all categories of preprocessing methods (like Ahlmann-Eltze and Huber (2023)), (ii) include a large, diverse set of real-world data (like Booeshaghi et al. (2022)), and (iii) use at least

the two following kinds of evaluations: "goal-driven" metrics that measure if the preprocessing produced normalized, variance-stabilized data (like Booeshaghi et al. (2022)), and "task-driven" metrics that measure influence of the preprocessing on downstream task performance, e.g., clustering or differential expression analysis. While such an effort was not in the scope of this thesis, it would be an exciting future work to advance the field of scRNA-seq data analysis, and to see whether our proposed Pearson residuals normalization framework will pass such a comprehensive test successfully.

For better scRNA-seq embedding methods, we need to put user experience first. With Chapter 5, we contributed to another ongoing benchmarking debate: how reliably can we visualize the high-dimensional gene expression space with 2D embeddings based on UMAP and t-SNE? Chari and Pachter (2023) claimed that these methods produce arbitrary embeddings, arguing that they preserved distances as well as a truly arbitrary elephant-shaped embedding. While we replicated the distance preservation results, we could show that UMAP and t-SNE performed very strongly and consistently on other relevant metrics like cluster and neighborhood preservation. Therefore, we could refute the claim that UMAP and t-SNE produce "arbitrary" embeddings. This set the records straight for many practitioners that were unsure if and how t-SNE and UMAP are still "safe to use" for scRNA-seq data.

What can we learn from this exchange of arguments? As in the discussion about benchmarks for normalization methods, the choice of metrics are at the heart of this debate. Choosing metrics means to decide which qualities of a method are important to us, and what we define as acceptable performance. For example, critics of UMAP and t-SNE have repeatedly argued that these methods are unacceptably bad, as their 2D embeddings only recalls 1–3 out of 10 nearest neighbors from high dimensions correctly. In contrast, one could argue that the other 7–9 nearest neighbors are not bad neighbors, because they are not random: Many of them are usually from a close vicinity of the correct neighborhood — in

scRNA-seq data often from the same cell type. Thus, we could interpret the same performance that the other side rejected as unacceptable as *soft* but successful neighborhood preservation (Figure 17).

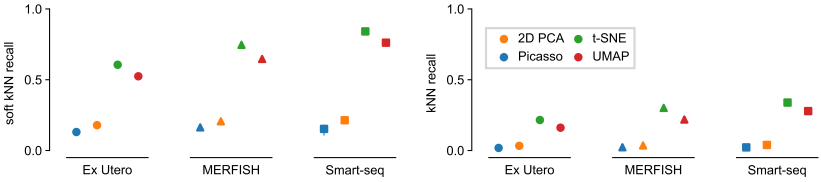

Figure 17: Soft vs. exact kNN recall. Left panel shows *soft* kNN recall, i.e., the fraction of the 10 nearest 2D neighbors that are among the 100 nearest neighbors in the original space. Right panel shows exact kNN recall as Figure 16d.

It is hard to say which perspective is correct here: Is it important for scRNA-seq analysis that kNN recall is exact, or is a soft notion of neighborhood preservation sufficient? The scRNA-seq field should try to address this kind of question for the many embedding quality metrics currently proposed (Espadoto et al., 2019; Sun et al., 2019; Heiser and Lau, 2020; Xiang et al., 2021), as they have more potential to move the field forward than asking once more "if the glass is half full or half empty".

That said, despite all benchmarking attempts, some fundamental limits to faithfully embedding high-dimensional data will remain. If we accept these limits, how can we still work with such an imperfect method in practice? One potentially helpful perspective here is that of the single-cell biologist who might apply UMAP or t-SNE without extended knowledge about high-dimensional space and the limits of dimensionality reduction. If they look at a typical 2D embedding with all its inevitable distortions — which wrong conclusions about the high-dimensional data will they draw automatically and involuntarily, simply based on how humans perceive visual inputs?

There is a long tradition of studying how humans extract information from scientific graphs (Quadri and Rosen, 2021), including work

on how distortions of various dimensionality reduction methods are perceived (Nonato and Aupetit, 2018). For example, Correll et al. (2018) tested which simple "data science sanity checks", like histograms or density plots, would mislead humans inspecting them. Similar studies on how 2D embeddings of scRNA-seq data are perceived by real humans could offer a new target to improve the user experience of these methods: Then, the criterion for a good "embedding user experience" would be whether or not a human observer will conclude something objectively wrong about the underlying high-dimensional data from it. Rather than trying to further improve the dimensionality reduction algorithms themselves, focus on the user experience would probably require work on which additional information the user needs to avoid falling for misleading distortions, i.e., embedding overlays that highlight suboptimally embedded points. First attempts to compute and communicate uncertainty estimates for embeddings positions already exist for the scRNA-seq field (Johnson et al., 2022; Xia et al., 2024), but these methods are still in their infancy, suffering, e.g., from poorly motivated null models.

Where will the atlases of scRNA-seq lead the neurosciences? The single-cell transcriptomics methods and atlas datasets presented in this thesis mapped out uncharted territory in the brain and brought many insights to the field of neuroscience — but what to do with the vast amounts of data?

The field of retina research has already shown a few possible directions (Shekhar and Sanes, 2021): Combining GWAS and scRNA-seq atlases allows to track where disease genes are active in the brain and might trigger disease onset (Roselli et al., 2018; Yi et al., 2021). Spatial transcriptomics allows to study, e.g., regional specialization in the retina (Choi et al., 2023) — and is only possible because of the numerous scRNA-seq atlas datasets that have been produced over the last decade. Multimodal approaches like Patch-seq allow connecting knowledge about transcriptomic cell types back to previously known morphological and functional cell types (Huang et al., 2022b). Cross-species scRNA-seq

atlas data comparisons already shed light on how mammalian nervous systems are related among each other (Hodge et al., 2019; Hahn et al., 2023) and with, e.g., reptile brains (Tosches et al., 2018). Also, more generally, scRNA-seq atlases are valuable because they define cell type markers, leading to much easier genetic access to these types. This will allow for targeted interventions and causal experiments on the role of cell types in specific circuits (He et al., 2016), e.g., with optogenetics.

These findings and those still to come have the potential to overturn the classical concept of brain cell types as discrete, static entities. Yuste et al. (2020) suggested that instead, the field needs a probabilistic cell type concept whose hierarchy takes into account the developmental and evolutionary origins of cell types (Arendt et al., 2019). Such a concept will hopefully be able to account for the continua between cell types that have been described in many brain regions already (Tasic et al., 2016, 2018; Harris et al., 2018). Currently, it is still unclear what these continua are good for — but it *is* clear that to find out, we need improved models and methods to process single-cell RNA sequencing data.

7 Author Contributions

Statement of contributions according to §9 (2): The presented thesis covers three projects and publications. The project descriptions and author contributions for each project are listed below. Contributions are described both as free text and according to the Contributor Roles Taxonomy (CRediT) system (Allen et al., 2019).

Analytic Pearson residuals
for normalization of single-cell RNA-seq UMI data
Lause, Berens & Kobak, *Genome Biology*, 2021

The project originated from investigating unexplained patterns in parameters of the scTransform method. We ended up simplifying the scTransform model to our analytic UMI count model, and ran computational experiments to demonstrate the simpler model works well as a data preprocessing pipeline. **Jan Lause** contributed analysis ideas, did literature research, ran all analyses, contributed the analysis code, and wrote and revised the manuscript draft. **Dmitry Kobak** contributed ideas, analysis directions, and manuscript feedback as the main supervisor. **Philipp Berens** discussed ideas for the method, contributed manuscript and analysis feedback and co-supervised the project.

CRediT: Conceptualization: JL, DK, PB; Methodology: JL, DK; Software: JL; Validation: JL; Formal analysis: JL; Investigation: JL, DK, PB; Resources: PB; Data Curation: JL; Writing - Original Draft: JL; Writing - Review & Editing: JL, DK, PB; Visualization: JL; Supervision: DK, PB; Project administration: DK, PB; Funding acquisition: PB

Compound models and Pearson residuals
for single-cell RNA-seq data without UMIs
Lause, Ziegenhain, Hartmanis, Berens & Kobak, *bioRxiv*, 2024

The project originated as a follow-up on the above project, and from ideas and discussions between Christoph Ziegenhain and Dmitry Kobak.

We extended the UMI count model to non-UMI read counts, using data from Ziegenhain & Hartmanis to constrain the model. Again, we ran computational experiments to demonstrate the model works well as a data preprocessing pipeline. **Jan Lause** contributed analysis ideas, derivations, ran all analyses, contributed the analysis code, and wrote and revised the manuscript draft. **Christoph Ziegenhain** contributed domain expertise, manuscript feedback, a reads-per-UMI dataset and code to reproduce that dataset. **Leonard Hartmanis** assisted in collecting that dataset. **Dmitry Kobak** contributed the modeling idea, analysis directions and manuscript feedback as the main supervisor. **Philipp Berens** discussed ideas for the method, contributed manuscript and analysis feedback and co-supervised the project.

CRediT: Conceptualization: JL, DK, PB; Methodology: JL, DK, CZ; Software: JL, CZ; Validation: JL; Formal analysis: JL; Investigation: JL, DK, CZ; Resources: PB, CZ; Data Curation: JL, CZ; Writing - Original Draft: JL; Writing - Review & Editing: JL, CZ, LH, DK, PB; Visualization: JL; Supervision: DK, PB; Project administration: DK, PB; Funding acquisition: PB

The art of seeing the elephant in the room:
2D embeddings of single-cell data do make sense
Lause, Berens & Kobak, *PLOS Computational Biology*, 2024

The project originated from a joint journal club on the Chari and Pachter (2023) paper, and a initial rebuttal analysis by Dmitry Kobak. We then extended the analysis to reproduce the Chari and Pachter (2023) results on scRNA-seq datasets. We also added new embedding quality metrics and a simulation experiment. **Jan Lause** ran the analysis, selected new metrics, contributed all analysis code, and wrote and revised the manuscript. **Dmitry Kobak** ran the initial pilot analysis and contributed feedback on analysis and manuscript as the main supervisor. **Philipp Berens** contributed manuscript and analysis feedback and co-supervised the project.

CRediT: Conceptualization: JL, DK; Methodology: JL, DK; Software: JL; Validation: JL; Formal analysis: JL; Investigation: JL, PB, DK; Resources: PB; Data Curation: JL; Writing - Original Draft: JL; Writing - Review & Editing: JL, DK, PB; Visualization: JL; Supervision: DK, PB; Project administration: DK, PB; Funding acquisition: PB

8 Acknowledgements

ten thousand hours
felt like ten thousand hands
ten thousand hands
they carry me

Macklemore

I have learned at least one thing in the last years working on this thesis: The P in PhD is for people — people that I want to thank for helping me get through this incredibly exciting, sometimes challenging, but so so joyful and fun time. I came a long way, and it would not be the same without you. Thank you all so much!

I want to thank Philipp, Thomas and Dmitry for their advice and guidance in academia and beyond — that includes career walks, card game nights and coin collection tricks!

Thanks to Philipp for the opportunity to explore science and academia as a friendly place with a path for everyone—maybe even me!

Thanks to Dmitry, for your explanations, ideas and patience.

Thanks everyone in the Berenslab — for lunch breaks, coffee breaks, cold brew, and for hot debates on social justice and olive oil.

Thanks to Thomas and the Eulerlab — for meetings, projects and parties!

Special thanks to Lara, for trips on boats and bikes and the connection to Aachen and Azeroth.

Thanks to Jonathan, for trips with tents at lakes.

Thanks to Rita, for coffee breaks and counseling.

Big thanks to Lisa, for helping me push this very document over the finish line — PhDone!

Thanks to Kyra, Sebastian, Lara, Nik and Lisa, for last-minute ideas and finding many typos in here. Thanks to Jonas, Fabio and Rita — for NDS! Thanks Ziwei — for Murakami. Thanks to Sarah Strauss, Sophie, Cornelius — for being around from the beginning.

Finally, and most dearly, to Pia, my family, and all my friends: Thanks for being with me, and for being who you are.

It's not the molecules. It's you!

References

RM Adelson. Compound Poisson distributions. *Journal of the Operational Research Society*, 17(1):73–75, 1966.

Abhi Aggarwal, Rui Liu, Yang Chen, Amelia J Ralowicz, Samuel J Bergerson, Filip Tomaska, Boaz Mohar, Timothy L Hanson, Jeremy P Hasseman, Daniel Reep, et al. Glutamate indicators with improved activation kinetics and localization for imaging synaptic transmission. *Nature Methods*, pages 1–10, 2023.

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *8th International Conference Database Theory, ICDT 2001: London, UK, January 4–6, 2001, Proceedings 8*, pages 420–434. Springer, 2001.

Alan Agresti. *Foundations of linear and generalized linear models*. John Wiley & Sons, 2015.

Alejandro Aguilera-Castrejon, Bernardo Oldak, Tom Shani, Nadir Ghanem, Chen Itzkovich, Sharon Slomovich, Shadi Tarazi, Jonathan Bayerl, Valeriya Chugaeva, Muneef Ayyash, et al. Ex utero mouse embryogenesis from pre-gastrulation to late organogenesis. *Nature*, 593(7857):119–124, 2021.

Constantin Ahlmann-Eltze and Wolfgang Huber. glmGamPoi: fitting Gamma-Poisson generalized linear models on single cell count data. *Bioinformatics*, 36(24):5701–5702, 2020.

Constantin Ahlmann-Eltze and Wolfgang Huber. Comparison of transformations for single-cell RNA-seq data. *Nature Methods*, 20(5):665–672, 2023.

Liz Allen, Alison O'Connell, and Veronique Kiermer. How can we ensure visibility and diversity in research contributions? how the Contributor Role Taxonomy (CRediT) is helping the shift from authorship to contributorship. *Learned Publishing*, 32(1):71–74, 2019.

Naomi Altman and Martin Krzywinski. The curse(s) of dimensionality. *Nature Methods*, 15(6):399–400, 2018.

Seth A Ament and Alexandros Poulopoulos. The brain's dark transcriptome: Sequencing RNA in distal compartments of neurons and glia. *Current Opinion in Neurobiology*, 81:102725, 2023.

A Ames III and FB Nesbett. In vitro retina as an experimental model of the central nervous system. *Journal of Neurochemistry*, 37(4):867–877, 1981.

Robert A Amezquita, Aaron TL Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, et al. Orchestrating single-cell analysis with Bioconductor. *Nature Methods*, 17(2):137–145, 2020.

Robert A Amezquita, Aaron TL Lun, Stephanie Hicks, Raphael Gottardo, and Peter Hickey. Basics of single-cell analysis with Bioconductor, 2024. URL `https://bioconductor.org/books/3.19/OSCA.basic/`.

Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):1–12, 2010.

Tallulah S Andrews, Vladimir Yu Kiselev, Davis McCarthy, and Martin Hemberg. Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. *Nature Protocols*, 16(1):1–9, 2021.

Francis J Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.

Detlev Arendt, Paola Yanina Bertucci, Kaia Achim, and Jacob M Musser. Evolution of neuronal types and families. *Current Opinion in Neurobiology*, 56:144–152, 2019.

Sanjeev Arora, Wei Hu, and Pravesh K Kothari. An analysis of the t-SNE algorithm for data visualization. In *Conference on Learning Theory*, pages 1455–1462. PMLR, 2018.

M Madan Babu, Nicholas M Luscombe, L Aravind, Mark Gerstein, and Sarah A Teichmann. Structure and evolution of transcriptional regulatory networks. *Current Opinion in Structural Biology*, 14(3):283–291, 2004.

Rhonda Bacher, Li-Fang Chu, Ning Leng, Audrey P Gasch, James A Thomson, Ron M Stewart, Michael Newton, and Christina Kendziorski. SCnorm: robust normalization of single-cell RNA-seq data. *Nature Methods*, 14(6):584–586, 2017.

Tom Baden, Philipp Berens, Katrin Franke, Miroslav Román Rosón, Matthias Bethge, and Thomas Euler. The functional diversity of retinal ganglion cells in the mouse. *Nature*, 529(7586):345–350, 2016.

Tom Baden, Thomas Euler, and Philipp Berens. Understanding the retinal basis of vision across species. *Nature Reviews Neuroscience*, 21 (1):5–20, 2020.

Shaul K Bar-Lev and Peter Enis. On the classical choice of variance stabilizing transformations and an application for a Poisson variate. *Biometrika*, 75(4):803–804, 1988.

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, 2019.

Katharine Best, Theres Oakes, James M Heather, John Shawe-Taylor, and Benny Chain. Computational analysis of stochastic heterogeneity in PCR amplification efficiency revealed by single molecule barcoding. *Scientific Reports*, 5(1):1–13, 2015.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. A unifying perspective on neighbor embeddings along the attraction-repulsion spectrum. *arXiv preprint arXiv:2007.08902*, 2020.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction-repulsion spectrum in neighbor embeddings. *The Journal of Machine Learning Research*, 23(1):4118–4149, 2022.

A Sina Booeshaghi, Ingileif B Hallgrímsdóttir, Ángel Gálvez-Merchán, and Lior Pachter. Depth normalization for single-cell genomics count data. *BioRxiv*, pages 2022–05, 2022.

George EP Box. Robustness in the strategy of scientific model building. In *Robustness in statistics*, pages 201–236. Elsevier, 1979.

Jérémie Breda, Mihaela Zavolan, and Erik van Nimwegen. Bayesian inference of gene expression states from single-cell RNA-seq data. *Nature Biotechnology*, 39(8):1008–1016, 2021.

Philip Brennecke, Simon Anders, Jong Kyoung Kim, Aleksandra A Kołodziejczyk, Xiuwei Zhang, Valentina Proserpio, Bianka Baying, Vladimir Benes, Sarah A Teichmann, John C Marioni, et al. Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, 10(11):1093–1095, 2013.

Jason D Buenrostro, Beijing Wu, Ulrike M Litzenburger, Dave Ruff, Michael L Gonzales, Michael P Snyder, Howard Y Chang, and William J Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, 2015.

T Tony Cai and Rong Ma. Theoretical foundations of t-SNE for visualizing high-dimensional clustered data. *Journal of Machine Learning Research*, 23(301):1–54, 2022.

Iván J Cajigas, Georgi Tushev, Tristan J Will, Susanne tom Dieck, Nicole Fuerst, and Erin M Schuman. The local transcriptome in the synaptic neuropil revealed by deep sequencing and high-resolution imaging. *Neuron*, 74(3):453–466, 2012.

Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M Ibrahim, Andrew J Hill, Fan Zhang, Stefan Mundlos, Lena Chris-

tiansen, Frank J Steemers, et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, 2019.

Yingying Cao, Simo Kitanovski, Ralf Küppers, and Daniel Hoffmann. UMI or not UMI, that is the question for scRNA-seq zero-inflation. *Nature Biotechnology*, 39(2):158–159, 2021.

Dorothée Chabas, Sergio E Baranzini, Dennis Mitchell, Claude CA Bernard, Susan R Rittling, David T Denhardt, Raymond A Sobel, Christopher Lock, Marcela Karpuj, Rosetta Pedotti, et al. The influence of the proinflammatory cytokine, osteopontin, on autoimmune demyelinating disease. *Science*, 294(5547):1731–1735, 2001.

Tara Chari and Lior Pachter. The specious art of single-cell genomics. *PLOS Computational Biology*, 19(8):e1011288, 2023.

Wenan Chen, Yan Li, John Easton, David Finkelstein, Gang Wu, and Xiang Chen. UMI-count modeling and differential expression analysis for single-cell RNA sequencing. *Genome Biology*, 19(1):1–17, 2018.

Jongsu Choi, Jin Li, Salma Ferdous, Qingnan Liang, Jeffrey R Moffitt, and Rui Chen. Spatial organization of the mouse retina at single cell resolution by MERFISH. *Nature Communications*, 14(1):4929, 2023.

Saket Choudhary and Rahul Satija. Comparison and evaluation of statistical error models for scRNA-seq. *bioRxiv*, 2021.

Neo Christopher Chung and John D Storey. Statistical significance of variables driving systematic variation in high-dimensional data. *Bioinformatics*, 31(4):545–554, 2014.

Brian S Clark, Genevieve L Stein-O'Brien, Fion Shiau, Gabrielle H Cannon, Emily Davis-Marcisak, Thomas Sherman, Clayton P Santiago, Thanh V Hoang, Fatemeh Rajaii, Rebecca E James-Esposito, et al. Single-cell RNA-seq analysis of retinal development identifies NFI factors as regulating mitotic exit and late-born cell specification. *Neuron*, 102(6):1111–1126, 2019.

Suzanne J Clark and Joe N Perry. Estimation of the negative binomial parameter κ by maximum quasi-likelihood. *Biometrics*, pages 309–316, 1989.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

Michael B Cole, Davide Risso, Allon Wagner, David DeTomaso, John Ngai, Elizabeth Purdom, Sandrine Dudoit, and Nir Yosef. Performance assessment and selection of normalization procedures for single-cell RNA-seq. *Cell Systems*, 8(4):315–328, 2019.

Michael Correll, Mingwei Li, Gordon Kindlmann, and Carlos Scheidegger. Looks good to me: Visualizations as sanity checks. *IEEE transactions on visualization and computer graphics*, 25(1):830–839, 2018.

Aedin Culhane. Correspondence analysis in R. `https://aedin.github.io/PCAworkshop/articles/c_COA.html`, 2020.

Fiona Cunningham, James E Allen, Jamie Allen, Jorge Alvarez-Jarreta, M Ridwan Amode, Irina M Armean, Olanrewaju Austine-Orimoloye, Andrey G Azov, If Barnes, Ruth Bennett, et al. Ensembl 2022. *Nucleic Acids Research*, 50(D1):D988–D995, 2022.

Sebastian Damrich, Manuel V Klockow, Philipp Berens, Fred A Hamprecht, and Dmitry Kobak. Visualizing single-cell data with the neighbor embedding spectrum. *bioRxiv*, pages 2024–04, 2024.

Spyros Darmanis, Caroline Julie Gallant, Voichita Dana Marinescu, Mia Niklasson, Anna Segerman, Georgios Flamourakis, Simon Fredriksson, Erika Assarsson, Martin Lundberg, Sven Nelander, et al. Simultaneous multiplexed measurement of RNA and proteins in single cells. *Cell Reports*, 14(2):380–389, 2016.

Winfried Denk and Heinz Horstmann. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biology*, 2(11):e329, 2004.

Bo Ding, Lina Zheng, Yun Zhu, Nan Li, Haiyang Jia, Rizi Ai, Andre Wildberg, and Wei Wang. Normalization and noise reduction for single cell RNA-seq experiments. *Bioinformatics*, 31(13):2225–2227, 2015.

Jiarui Ding, Xian Adiconis, Sean K Simmons, Monika S Kowalczyk, Cynthia C Hession, Nemanja D Marjanovic, Travis K Hughes, Marc H Wadsworth, Tyler Burks, Lan T Nguyen, et al. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology*, 38(6):737–746, 2020.

Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

RA Dorfman. A note on the! d-method for finding variance formulae. *Biometric Bulletin*, 1938.

John E Dowling. *The retina: an approachable part of the brain*. Harvard University Press, 1987.

Angelo Duò, Mark D Robinson, and Charlotte Soneson. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, 7, 2018.

Joseph R Ecker, Daniel H Geschwind, Arnold R Kriegstein, John Ngai, Pavel Osten, Damon Polioudakis, Aviv Regev, Nenad Sestan, Ian R Wickersham, and Hongkui Zeng. The BRAIN initiative cell census consortium: lessons learned toward generating a comprehensive brain cell atlas. *Neuron*, 96(3):542–557, 2017.

Nils Eling, Arianne C Richard, Sylvia Richardson, John C Marioni, and Catalina A Vallejos. Correcting the mean-variance dependency for

differential variability testing using single-cell RNA sequencing data. *Cell Systems*, 7(3):284–294, 2018.

Mateus Espadoto, Rafael M Martins, Andreas Kerren, Nina ST Hirata, and Alexandru C Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2153–2173, 2019.

William Feller. On a general class of 'contagious' distributions. *The Annals of Mathematical Statistics*, 14(4):389–400, 1943.

Huijuan Feng, Daniel F Moakley, Shuonan Chen, Melissa G McKenzie, Vilas Menon, and Chaolin Zhang. Complexity and graded regulation of neuronal cell-type–specific alternative splicing revealed by single-cell RNA sequencing. *Proceedings of the National Academy of Sciences*, 118(10):e2013056118, 2021.

Greg Finak, Andrew McDavid, Masanao Yajima, Jingyuan Deng, Vivian Gersuk, Alex K Shalek, Chloe K Slichter, Hannah W Miller, M Juliana McElrath, Martin Prlic, et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16:1–13, 2015.

Katrin Franke, Philipp Berens, Timm Schubert, Matthias Bethge, Thomas Euler, and Tom Baden. Inhibition decorrelates visual feature representations in the inner retina. *Nature*, 542(7642):439–444, 2017.

Murray F Freeman and John W Tukey. Transformations related to the angular and the square root. *The Annals of Mathematical Statistics*, pages 607–611, 1950.

Daniel R Garalde, Elizabeth A Snell, Daniel Jachimowicz, Botond Sipos, Joseph H Lloyd, Mark Bruce, Nadia Pantic, Tigist Admassu, Phillip James, Anthony Warland, et al. Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods*, 15(3):201–206, 2018.

Charles Gawad, Winston Koh, and Stephen R Quake. Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics*, 17 (3):175–188, 2016.

Michael Greenacre. *Correspondence analysis in practice*. Chapman and Hall/CRC, 2007.

Michael Greenacre and Trevor Hastie. The geometric interpretation of correspondence analysis. *Journal of the American Statistical Association*, 82(398):437–447, 1987.

Dominic Grün. Revealing dynamics of gene expression variability in cell state space. *Nature Methods*, 17(1):45–49, 2020.

Dominic Grün, Lennart Kester, and Alexander Van Oudenaarden. Validation of noise models for single-cell transcriptomics. *Nature Methods*, 11(6):637–640, 2014.

Christoph Hafemeister and Rahul Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20(1):1–15, 2019.

Michael Hagemann-Jensen, Christoph Ziegenhain, Ping Chen, Daniel Ramsköld, Gert-Jan Hendriks, Anton JM Larsson, Omid R Faridani, and Rickard Sandberg. Single-cell RNA counting at allele and isoform resolution using Smart-seq3. *Nature Biotechnology*, 38(6):708–714, 2020.

Michael Hagemann-Jensen, Christoph Ziegenhain, and Rickard Sandberg. Scalable single-cell RNA sequencing from full transcripts with Smart-seq3xpress. *Nature Biotechnology*, 40(10):1452–1457, 2022.

Joshua Hahn, Aboozar Monavarfeshani, Mu Qiao, Allison H Kao, Yvonne Kölsch, Ayush Kumar, Vincent P Kunze, Ashley M Rasys, Rose Richardson, Joseph B Wekselblatt, et al. Evolution of neuronal cell classes and types in the vertebrate retina. *Nature*, 624(7991): 415–424, 2023.

Xiaoping Han, Renying Wang, Yincong Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Fang Ye, et al. Mapping the mouse cell atlas by microwell-seq. *Cell*, 172(5): 1091–1107, 2018.

Yuchen Hao and Andrew JR Plested. Seeing glutamate at central synapses. *Journal of Neuroscience Methods*, 375:109531, 2022.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

Kenneth D Harris, Hannah Hochgerner, Nathan G Skene, Lorenza Magno, Linda Katona, Carolina Bengtsson Gonzales, Peter Somogyi, Nicoletta Kessaris, Sten Linnarsson, and Jens Hjerling-Leffler. Classes and continua of hippocampal CA1 inhibitory neurons revealed by single-cell transcriptomics. *PLoS Biology*, 16(6):e2006387, 2018.

Miao He, Jason Tucciarone, SooHyun Lee, Maximiliano José Nigro, Yongsoo Kim, Jesse Maurica Levine, Sean Michael Kelly, Illya Krugikov, Priscilla Wu, Yang Chen, et al. Strategies and tools for combinatorial targeting of GABAergic neurons in mouse cerebral cortex. *Neuron*, 91(6):1228–1243, 2016.

Graham Heimberg, Rajat Bhatnagar, Hana El-Samad, and Matt Thomson. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell systems*, 2(4):239–250, 2016.

Cody N Heiser and Ken S Lau. A quantitative framework for evaluating single-cell data structure preservation by dimensionality reduction techniques. *Cell Reports*, 31(5), 2020.

Moritz Helmstaedter, Kevin L Briggman, Srinivas C Turaga, Viren Jain, H Sebastian Seung, and Winfried Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461): 168–174, 2013.

Lukas Heumos, Anna C Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D Lücken, Daniel C Strobl, Juan Henao, Fabiola Curion, et al. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8):550–572, 2023.

Mark O Hill. Correspondence analysis: a neglected multivariate method. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 23(3):340–354, 1974.

Hermann O Hirschfeld. A connection between correlation and contingency. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 520–524. Cambridge University Press, 1935.

Rebecca D Hodge, Trygve E Bakken, Jeremy A Miller, Kimberly A Smith, Eliza R Barkan, Lucas T Graybuck, Jennie L Close, Brian Long, Nelson Johansen, Osnat Penn, et al. Conserved cell types with divergent features in human versus mouse cortex. *Nature*, 573(7772): 61–68, 2019.

Susan Holmes. Multivariate data analysis: the French way. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 219–233. Institute of Mathematical Statistics, 2008.

Guosong Hong and Charles M Lieber. Novel electrode technologies for neural recordings. *Nature Reviews Neuroscience*, 20(6):330–345, 2019.

Lauren L Hsu and Aedín C Culhane. Correspondence analysis for dimension reduction, batch integration, and visualization of single-cell RNA-seq data. *Scientific Reports*, 13(1):1–17, 2023.

Haiyang Huang, Yingfan Wang, Cynthia Rudin, and Edward P Browne. Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization. *Communications Biology*, 5(1): 719, 2022a.

Wanjing Huang, Qiang Xu, Jing Su, Lei Tang, Zhao-Zhe Hao, Chuan Xu, Ruifeng Liu, Yuhui Shen, Xuan Sang, Nana Xu, et al. Linking transcriptomes with morphological and functional phenotypes in mammalian retinal ganglion cells. *Cell Reports*, 40(11), 2022b.

Wolfgang Huber, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, et al. Orchestrating high-throughput genomic analysis with bioconductor. *Nature methods*, 12 (2):115–121, 2015.

J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & Molecular Medicine*, 50(8):1–14, 2018.

H Hyden. Behavior, neural function, and RNA. *Progress in nucleic acid research and molecular biology*, 6:187–218, 1967.

Rafael Irizarry. R package with methods for small counts stored in a sparse matrix. `https://github.com/rafalab/smallcount`, 2021.

Saiful Islam, Amit Zeisel, Simon Joost, Gioele La Manno, Pawel Zajac, Maria Kasper, Peter Lönnerberg, and Sten Linnarsson. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163–166, 2014.

Lichun Jiang, Felix Schlesinger, Carrie A Davis, Yu Zhang, Renhua Li, Marc Salit, Thomas R Gingeras, and Brian Oliver. Synthetic spike-in standards for RNA-seq experiments. *Genome Research*, 21(9):1543–1551, 2011.

Gudlaugur Jóhannesson, Gunnlaugur Björnsson, and Einar H Gudmundsson. Afterglow light curves and broken power laws: a statistical study. *The Astrophysical Journal*, 640(1):L5, 2006.

Eric M Johnson, William Kath, and Madhav Mani. EMBEDR: distinguishing signal from noise in single-cell omics data. *Patterns*, 3(3), 2022.

Norman L Johnson, Adrienne W Kemp, and Samuel Kotz. *Univariate discrete distributions*, volume 444. John Wiley & Sons, 2005.

Per Johnsson, Christoph Ziegenhain, Leonard Hartmanis, Gert-Jan Hendriks, Michael Hagemann-Jensen, Björn Reinius, and Rickard Sandberg. Transcriptional kinetics and molecular functions of long noncoding RNAs. *Nature Genetics*, 54(3):306–317, 2022.

James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

Ino D Karemaker and Michiel Vermeulen. Single-cell DNA methylation profiling: technologies and biological applications. *Trends in Biotechnology*, 36(9):952–965, 2018.

Yukie Kashima, Yoshitaka Sakamoto, Keiya Kaneko, Masahide Seki, Yutaka Suzuki, and Ayako Suzuki. Single-cell sequencing techniques from individual to multiomics analyses. *Experimental & Molecular Medicine*, 52(9):1419–1427, 2020.

Nobutaka Kawahara, Yan Wang, Akitake Mukasa, Kazuhide Furuya, Tatsuya Shimizu, Takao Hamakubo, Hiroyuki Aburatani, Tatsuhiko Kodama, and Takaaki Kirino. Genome-wide gene expression analysis for induced ischemic tolerance and delayed neuronal death following transient global ischemia in rats. *Journal of Cerebral Blood Flow & Metabolism*, 24(2):212–233, 2004.

C. D. Kemp. 'Stuttering-Poisson' distributions. *Journal of the Statistical and Social Inquiry Society of Ireland*, XXI(V):151–157, May 1967.

Peter V Kharchenko. The triumphs and limitations of computational methods for scRNA-seq. *Nature Methods*, 18(7):723–732, 2021.

Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742, 2014.

Dong-Wook Kim, Zizhen Yao, Lucas T Graybuck, Tae Kyung Kim, Thuc Nghi Nguyen, Kimberly A Smith, Olivia Fong, Lynn Yi, Noushin Koulena, Nico Pierson, et al. Multimodal analysis of cell types in a hypothalamic node controlling social behavior. *Cell*, 179(3):713–728, 2019.

Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Tomislav Ilicic, Sarah A Teichmann, and John C Marioni. Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nature Communications*, 6(1):1–9, 2015.

Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015.

Sandy L Klemm, Zohar Shipony, and William J Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220, 2019.

Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1):5416, 2019.

Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann. The technology and biology of single-cell RNA sequencing. *Molecular Cell*, 58(4):610–620, 2015.

Antonis Koussounadis, Simon P Langdon, In Hwa Um, David J Harrison, and V Anne Smith. Relationship between differentially expressed mRNA and mRNA-protein correlations in a xenograft model system. *Scientific reports*, 5(1):10775, 2015.

Rishikesh U Kulkarni and Evan W Miller. Voltage imaging: pitfalls and potential. *Biochemistry*, 56(39):5171–5177, 2017.

Jan Lause. Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. `https://github.com/berenslab/umi-normalization`, 2021.

Jan Lause, Philipp Berens, and Dmitry Kobak. Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. *Genome Biology*, 22(1):1–20, 2021.

Jan Lause, Christoph Ziegenhain, Leonard Hartmanis, Philipp Berens, and Dmitry Kobak. Compound models and pearson residuals for normalization of single-cell RNA-seq data without UMIs. *bioRxiv*, 2023.

Jan Lause, Philipp Berens, and Dmitry Kobak. The art of seeing the elephant in the room: 2D embeddings of single-cell data do make sense. *PLOS Computational Biology*, 20(10):1–5, 2024.

Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, 15(2):1–17, 2014.

John A Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7-9):1431–1443, 2009.

Bo Li and Colin N Dewey. RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinformatics*, 12:1–16, 2011.

Bo Li, Victor Ruotti, Ron M Stewart, James A Thomson, and Colin N Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500, 2010.

Michael Z Lin and Mark J Schnitzer. Genetically encoded indicators of neuronal activity. *Nature Neuroscience*, 19(9):1142–1153, 2016.

George C Linderman and Stefan Steinerberger. Clustering with t-SNE, provably. *arXiv*, pages arXiv–1706, 2017.

George C Linderman and Stefan Steinerberger. Clustering with t-SNE, provably. *SIAM Journal on Mathematics of Data Science*, 1(2):313–332, 2019.

George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nature Methods*, 16 (3):243–245, 2019.

Marcela Lipovsek, Cedric Bardy, Cathryn R Cadwell, Kristen Hadley, Dmitry Kobak, and Shreejoy J Tripathy. Patch-seq: Past, present, and future. *Journal of Neuroscience*, 41(5):937–946, 2021.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018.

Lucille Lopez-Delisle and Jean-Baptiste Delisle. baredSC: Bayesian approach to retrieve expression distribution of single-cell data. *BMC Bioinformatics*, 23(1):36, 2022.

Dominique Lord. Modeling motor vehicle crashes using Poisson-gamma models: Examining the effects of low sample mean values and small

sample size on the estimation of the fixed dispersion parameter. *Accident Analysis & Prevention*, 38(4):751–766, 2006.

Dominique Lord and Luis F Miranda-Moreno. Effects of low sample mean values and small sample size on the estimation of the fixed dispersion parameter of Poisson-gamma models for modeling motor vehicle crashes: A Bayesian perspective. *Safety Science*, 46(5):751–770, 2008.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):1–21, 2014.

Malte D Luecken and Fabian J Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6): e8746, 2019.

Aaron TL Lun. Overcoming systematic errors caused by log-transformation of normalized single-cell RNA sequencing data. *bioRxiv*, page 404962, 2018.

Aaron TL Lun, Karsten Bach, and John C Marioni. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, 17(1):1–14, 2016.

Aaron TL Lun, Fernando J Calero-Nieto, Liora Haim-Vilmovsky, Berthold Göttgens, and John C Marioni. Assessing the reliability of spike-in normalization for analyses of single-cell RNA sequencing data. *Genome research*, 27(11):1795–1806, 2017.

Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5): 1202–1214, 2015.

Richard H Masland. The neuronal organization of the retina. *Neuron*, 76(2):266–280, 2012.

Hansruedi Mathys, Carles A Boix, Leyla Anne Akay, Ziting Xia, Jose Davila-Velderrain, Ayesha P Ng, Xueqiao Jiang, Ghada Abdelhady, Kyriaki Galani, Julio Mantero, et al. Single-cell multiregion dissection of Alzheimer's disease. *Nature*, pages 1–11, 2024.

Andrew Matus. Microtubule-associated proteins: their potential role in determining neuronal morphology. *Annual Review of Beuroscience*, 11 (1):29–44, 1988.

Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018.

MICrONS Consortium, J Alexander Bae, Mahaly Baptiste, Caitlyn A Bishop, Agnes L Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J Bumbarger, Manuel A Castro, Brendan Celii, et al. Functional connectomics spanning multiple areas of mouse visual cortex. *BioRxiv*, pages 2021–07, 2021.

Kevin R Moon, Jay S Stanley III, Daniel Burkhardt, David van Dijk, Guy Wolf, and Smita Krishnaswamy. Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018.

TS Moothathu and C Satheesh Kumar. Some properties of the stuttering Poisson distribution. *Calcutta Statistical Association Bulletin*, 45(1-2):125–130, 1995.

Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods*, 5(7):621–628, 2008.

Anna Neufeld, Joshua Popp, Lucy L Gao, Alexis Battle, and Daniela Witten. Negative binomial count splitting for single-cell RNA sequencing data. *arXiv*, 2023.

Luis Gustavo Nonato and Michael Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2018.

Dimitry Ofengeim, Nikolaos Giagtzoglou, Dann Huh, Chengyu Zou, and Junying Yuan. Single-cell RNA sequencing: unraveling the brain one cell at a time. *Trends in Molecular Medicine*, 23(6):563–576, 2017.

Lior Pachter, 2021. URL `https://web.archive.org/web/20240729115631/https://archive.is/2024.07.29-115414/https://x.com/lpachter/status/1431325969411821572`.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Aleksandra A Petelski, Edward Emmott, Andrew Leduc, R Gray Huffman, Harrison Specht, David H Perlman, and Nikolai Slavov. Multiplexed single-cell proteomics using SCoPE2. *Nature Protocols*, 16(12): 5398–5425, 2021.

Belinda Phipson, Luke Zappia, and Alicia Oshlack. Gene length and detection bias in single cell RNA sequencing protocols. *F1000Research*, 6, 2017.

Simone Picelli, Åsa K Björklund, Omid R Faridani, Sven Sagasser, Gösta Winberg, and Rickard Sandberg. Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nature Methods*, 10(11):1096–1098, 2013.

Emma Pierson and Christopher Yau. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(1):1–10, 2015.

Pavlin G Poličar, Martin Stražar, and Blaž Zupan. openTSNE: a modular python library for t-SNE dimensionality reduction and embedding. *bioRxiv*, page 731877, 2019.

Pavlin G Poličar, Martin Stražar, and Blaž Zupan. openTSNE: A modular python library for t-SNE dimensionality reduction and embedding. *Journal of Statistical Software*, 109:1–30, 2024.

Ashok Prasad and Elaheh Alizadeh. Cell form and function: interpreting and controlling the shape of adherent cells. *Trends in Biotechnology*, 37(4):347–357, 2019.

Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. Single-cell mRNA quantification and differential analysis with Census. *Nature Methods*, 14(3):309–315, 2017.

Ghulam Jilani Quadri and Paul Rosen. A survey of perception-based visualization studies by task. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):5026–5048, 2021.

Arjun Raj, Charles S Peskin, Daniel Tranchina, Diana Y Vargas, and Sanjay Tyagi. Stochastic mRNA synthesis in mammalian cells. *PLoS Biology*, 4(10):e309, 2006.

Daniel Ramsköld, Shujun Luo, Yu-Chieh Wang, Robin Li, Qiaolin Deng, Omid R Faridani, Gregory A Daniels, Irina Khrebtukova, Jeanne F Loring, Louise C Laurent, et al. Full-length mRNA-Seq from single-cell

levels of RNA and individual circulating tumor cells. *Nature Biotechnology*, 30(8):777–782, 2012.

Jorge S Reis-Filho. Next-generation sequencing. *Breast Cancer Research*, 11(Suppl 3):S12, 2009.

Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications*, 9 (1):284, 2018.

Reyna Edith Rosales-Alvarez, Jasmin Rettkowski, Josip Stefan Herman, Gabrijela Dumbović, Nina Cabezas-Wallscheid, and Dominic Grün. VarID2 quantifies gene expression noise dynamics and unveils functional heterogeneity of ageing hematopoietic stem cells. *Genome Biology*, 24(1):1–30, 2023.

Carolina Roselli, Mark D Chaffin, Lu-Chen Weng, Stefanie Aeschbacher, Gustav Ahlberg, Christine M Albert, Peter Almgren, Alvaro Alonso, Christopher D Anderson, Krishna G Aragam, et al. Multi-ethnic genome-wide association study for atrial fibrillation. *Nature Genetics*, 50(9):1225–1233, 2018.

Assaf Rotem, Oren Ram, Noam Shoresh, Ralph A Sperling, Alon Goren, David A Weitz, and Bradley E Bernstein. Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state. *Nature Biotechnology*, 33(11):1165–1172, 2015.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

Abhishek Sarkar and Matthew Stephens. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature Genetics*, 53(6):770–777, 2021.

Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, 2015.

Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with Python. In *9th Python in Science Conference*, 2010.

Karthik Shekhar and Joshua R Sanes. Generating and using transcriptomically based retinal cell atlases. *Annual Review of Vision Science*, 7(1):43–72, 2021.

Karthik Shekhar, Sylvain W Lapan, Irene E Whitney, Nicholas M Tran, Evan Z Macosko, Monika Kowalczyk, Xian Adiconis, Joshua Z Levin, James Nemesh, Melissa Goldman, et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166(5): 1308–1323, 2016.

Amartya Singh and Hossein Khiabanian. Feature selection followed by a novel residuals-based normalization that includes variance stabilization simplifies and improves single-cell gene expression analysis. *BMC Bioinformatics*, 25, 2024.

Shaked Slovin, Annamaria Carissimo, Francesco Panariello, Antonio Grimaldi, Valentina Bouché, Gennaro Gambardella, and Davide Cacchiarelli. Single-cell RNA sequencing analysis: a step-by-step overview. *RNA Bioinformatics*, pages 343–365, 2021.

Harrison Specht, Edward Emmott, Aleksandra A Petelski, R Gray Huffman, David H Perlman, Marco Serra, Peter Kharchenko, Antonius Koller, and Nikolai Slavov. Single-cell proteomic and transcriptomic analysis of macrophage heterogeneity using SCoPE2. *Genome Biology*, 22(1):1–27, 2021.

Karpagam Srinivasan, Brad A Friedman, Jessica L Larson, Benjamin E Lauffer, Leonard D Goldstein, Laurie L Appling, Jovencio Borneo,

Chungkee Poon, Terence Ho, Fang Cai, et al. Untangling the brain's neuroinflammatory and neurodegenerative transcriptional responses. *Nature communications*, 7(1):11295, 2016.

Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145, 2015.

Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372 (6539):eabf4588, 2021.

Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, 14(9):865–868, 2017.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

Shiquan Sun, Jiaqiang Zhu, Ying Ma, and Xiang Zhou. Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. *Genome Biology*, 20:1–21, 2019.

Valentine Svensson. Droplet scRNA-seq is not zero-inflated. *Nature Biotechnology*, 38(2):147–150, 2020.

Valentine Svensson, Kedar Nath Natarajan, Lam-Ha Ly, Ricardo J Miragaia, Charlotte Labalette, Iain C Macaulay, Ana Cvejic, and Sarah A Teichmann. Power analysis of single-cell RNA-sequencing experiments. *Nature Methods*, 14(4):381–387, 2017.

Valentine Svensson, Eduardo da Veiga Beltrame, and Lior Pachter. A curated database reveals trends in single-cell transcriptomics. *Database*, 2020a.

Valentine Svensson, Adam Gayoso, Nir Yosef, and Lior Pachter. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics*, 2020b.

Wenhao Tang, François Bertaux, Philipp Thomas, Claire Stefanelli, Malika Saint, Samuel Marguerat, and Vahid Shahrezaei. bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics*, 36(4):1174–1181, 2020.

Bosiljka Tasic. Single cell transcriptomics in neuroscience: cell classification and beyond. *Current Opinion in Neurobiology*, 50:242–249, 2018.

Bosiljka Tasic, Vilas Menon, Thuc Nghi Nguyen, Tae Kyung Kim, Tim Jarsky, Zizhen Yao, Boaz Levi, Lucas T Gray, Staci A Sorensen, Tim Dolbeare, et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature Neuroscience*, 19(2):335–346, 2016.

Bosiljka Tasic, Zizhen Yao, Lucas T Graybuck, Kimberly A Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N Economo, Sarada Viswanathan, et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729): 72–78, 2018.

Luyi Tian, Xueyi Dong, Saskia Freytag, Kim-Anh Lê Cao, Shian Su, Abolfazl JalalAbadi, Daniela Amann-Zalcenstein, Tom S Weber, Azadeh Seidi, Jafar S Jabbari, et al. Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nature Methods*, 16(6):479–487, 2019.

Maria Antonietta Tosches, Tracy M Yamawaki, Robert K Naumann, Ariel A Jacobi, Georgi Tushev, and Gilles Laurent. Evolution of pallium, hippocampus, and cortical cell types revealed by single-cell transcriptomics in reptiles. *Science*, 360(6391):881–888, 2018.

F William Townes and Rafael A Irizarry. Quantile normalization of single-cell RNA-seq read counts without unique molecular identifiers. *Genome Biology*, 21(1):1–17, 2020.

F William Townes, Stephanie C Hicks, Martin J Aryee, and Rafael A Irizarry. Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biology*, 20:295, 2019.

Nicholas M Tran, Karthik Shekhar, Irene E Whitney, Anne Jacobi, Inbal Benhar, Guosong Hong, Wenjun Yan, Xian Adiconis, McKinzie E Arnold, Jung Min Lee, et al. Single-cell profiles of retinal ganglion cells differing in resilience to injury reveal neuroprotective genes. *Neuron*, 104(6):1039–1055, 2019.

James R Tribble, Asta Vasalauskaite, Tony Redmond, Robert D Young, Shoaib Hassan, Michael P Fautsch, Frank Sengpiel, Pete A Williams, and James E Morgan. Midget retinal ganglion cell dendritic and mitochondrial degeneration is an early feature of human glaucoma. *Brain Communications*, 1(1):fcz035, 2019.

Po-Yuan Tung, John D Blischak, Chiaowen Joyce Hsiao, David A Knowles, Jonathan E Burnett, Jonathan K Pritchard, and Yoav Gilad. Batch effects and the effective design of single-cell gene expression studies. *Scientific Reports*, 7(1):39921, 2017.

Catalina A Vallejos, John C Marioni, and Sylvia Richardson. BASiCS: Bayesian analysis of single-cell sequencing data. *PLoS Computational Biology*, 11(6):e1004333, 2015.

Catalina A Vallejos, Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C Marioni. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature Methods*, 14(6):565–571, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

Vincent Villette, Mariya Chavarha, Ivan K Dimov, Jonathan Bradley, Lagnajeet Pradhan, Benjamin Mathieu, Stephen W Evans, Simon Chamberland, Dongqing Shi, Renzhi Yang, et al. Ultrafast two-photon imaging of a high-gain voltage indicator in awake behaving mice. *Cell*, 179(7):1590–1608, 2019.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080, 2009.

Isaac Virshup, Sergei Rybakov, Fabian J Theis, Philipp Angerer, and F Alexander Wolf. anndata: Annotated data. *bioRxiv*, pages 2021–12, 2021.

Luke F Vistain and Savaş Tay. Single-cell proteomics. *Trends in Biochemical Sciences*, 46(8):661–672, 2021.

Christopher S Von Bartheld, Jami Bahney, and Suzana Herculano-Houzel. The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *Journal of Comparative Neurology*, 524(18):3865–3895, 2016.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

Trung Nghia Vu, Quin F Wills, Krishna R Kalari, Nifang Niu, Liewei Wang, Mattias Rantalainen, and Yudi Pawitan. Beta-Poisson model for single-cell RNA-seq data analyses. *Bioinformatics*, 32(14):2128–2135, 2016.

Florian Wagner. Straightforward clustering of single-cell RNA-Seq data with t-SNE and DBSCAN. *BioRxiv*, page 770388, 2019.

Florian Wagner. Monet: An open-source Python package for analyzing and integrating scRNA-Seq data using PCA-based latent spaces. *bioRxiv*, 2020.

Jingshu Wang, Mo Huang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, John Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. Gene expression distribution deconvolution in single-cell RNA sequencing. *Proceedings of the National Academy of Sciences*, 115(28):E6437–E6446, 2018.

Kaiwen Wang, Yuqiu Yang, Fangjiang Wu, Bing Song, Xinlei Wang, and Tao Wang. Comparative analysis of dimension reduction methods for cytometry by time-of-flight data. *Nature Communications*, 14(1):1836, 2023a.

Lingfei Wang. Single-cell normalization and association testing unifying CRISPR screen and gene co-expression analyses with Normalisr. *Nature Communications*, 12(1):6395, 2021.

Shu Wang, Eduardo D Sontag, and Douglas A Lauffenburger. What cannot be seen correctly in 2D visualizations of single-cell 'omics data? *Cell Systems*, 14(9):723–731, 2023b.

David I Warton. Why you cannot transform your way out of trouble for small counts. *Biometrics*, 74(1):362–368, 2018.

Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-SNE effectively. *Distill*, 1(10):e2, 2016.

Quin F Wills, Kenneth J Livak, Alex J Tipping, Tariq Enver, Andrew J Goldson, Darren W Sexton, and Chris Holmes. Single-cell gene expression analysis reveals genetic associations masked in whole-tissue experiments. *Nature Biotechnology*, 31(8):748–752, 2013.

LJ Willson, JL Folks, and JH Young. Complete sufficiency and maximum likelihood estimation for the two-parameter negative binomial distribution. *Metrika*, 33(1):349–362, 1986.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:1–5, 2018.

Chunlei Wu, Ian MacLeod, and Andrew I Su. BioGPS and MyGene.info: organizing online, gene-centric information. *Nucleic Acids Research*, 41(D1):D561–D565, 2013.

Zhijin Wu, Kenong Su, and Hao Wu. Non-linear normalization for non-UMI single cell RNA-Seq. *Frontiers in Genetics*, 12:612670, 2021.

Lucy Xia, Christy Lee, and Jingyi Jessica Li. Statistical method scDEED for detecting dubious 2D single-cell embeddings and optimizing t-SNE and UMAP hyperparameters. *Nature Communications*, 15(1):1753, 2024.

Ruizhi Xiang, Wencan Wang, Lei Yang, Shiyuan Wang, Chaohan Xu, and Xiaowen Chen. A comparison for dimensionality reduction methods of single-cell RNA-seq data. *Frontiers in Genetics*, 12:646936, 2021.

Wenjun Yan, Mallory A Laboulaye, Nicholas M Tran, Irene E Whitney, Inbal Benhar, and Joshua R Sanes. Mouse retinal cell atlas: molecular identification of over sixty amacrine cell types. *Journal of Neuroscience*, 40(27):5177–5195, 2020.

Zizhen Yao, Hanqing Liu, Fangming Xie, Stephan Fischer, Ricky S Adkins, Andrew I Aldridge, Seth A Ament, Anna Bartlett, M Margarita Behrens, Koen Van den Berge, et al. A transcriptomic and epigenomic cell atlas of the mouse primary motor cortex. *Nature*, 598(7879):103–110, 2021.

Zizhen Yao, Cindy TJ van Velthoven, Michael Kunst, Meng Zhang, Delissa McMillen, Changkyu Lee, Won Jung, Jeff Goldy, Aliya Abdelhak, Matthew Aitken, et al. A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain. *Nature*, 624(7991): 317–332, 2023.

Wenyang Yi, Yufeng Lu, Suijuan Zhong, Mei Zhang, Le Sun, Hao Dong, Mengdi Wang, Min Wei, Haohuan Xie, Hongqiang Qu, et al. A single-cell transcriptome atlas of the aging human and macaque retina. *National Science Review*, 8(4):nwaa179, 2021.

Wenjing Yin, Derrick Brittain, Jay Borseth, Marie E Scott, Derric Williams, Jedediah Perkins, Christopher S Own, Matthew Murfitt, Russel M Torres, Daniel Kapner, et al. A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy. *Nature Communications*, 11(1):4949, 2020.

Shun H Yip, Pak Chung Sham, and Junwen Wang. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Briefings in bioinformatics*, 20(4):1583–1589, 2019.

Rafael Yuste, Michael Hawrylycz, Nadia Aalling, Argel Aguilar-Valles, Detlev Arendt, Ruben Armañanzas, Giorgio A Ascoli, Concha Bielza, Vahid Bokharaie, Tobias Borgtoft Bergmann, et al. A community-based transcriptomics classification and nomenclature of neocortical cell types. *Nature Neuroscience*, 23(12):1456–1468, 2020.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, 18(1):174, 2017.

Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347 (6226):1138–1142, 2015.

Amit Zeisel, Hannah Hochgerner, Peter Lönnerberg, Anna Johnsson, Fatima Memic, Job Van Der Zwan, Martin Häring, Emelie Braun, Lars E Borm, Gioele La Manno, et al. Molecular architecture of the mouse nervous system. *Cell*, 174(4):999–1014, 2018.

Hongkui Zeng and Joshua R Sanes. Neuronal cell-type classification: challenges, opportunities and the path forward. *Nature Reviews Neuroscience*, 18(9):530–546, 2017.

Meng Zhang, Stephen W Eichhorn, Brian Zingg, Zizhen Yao, Kaelan Cotter, Hongkui Zeng, Hongwei Dong, and Xiaowei Zhuang. Spatially resolved cell atlas of the mouse primary motor cortex by merfish. *Nature*, 598(7879):137–143, 2021.

Yan Zhang, Márton Rózsa, Yajie Liang, Daniel Bushey, Ziqiang Wei, Jihong Zheng, Daniel Reep, Gerard Joey Broussard, Arthur Tsang, Getahun Tsegaye, et al. Fast and sensitive GCaMP calcium indicators for imaging neural populations. *Nature*, 615(7954):884–891, 2023.

Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):1–12, 2017.

Christoph Ziegenhain, Beate Vieth, Swati Parekh, Björn Reinius, Amy Guillaumet-Adkins, Martha Smets, Heinrich Leonhardt, Holger Heyn, Ines Hellmann, and Wolfgang Enard. Comparative analysis of single-cell RNA sequencing methods. *Molecular Cell*, 65(4):631–643, 2017.

Christoph Ziegenhain, Gert-Jan Hendriks, Michael Hagemann-Jensen, and Rickard Sandberg. Molecular spikes: a gold standard for single-cell RNA counting. *Nature Methods*, 19(5):560–566, 2022.

9 Supplementary Figures

9.1 Analytic Pearson residuals for UMI counts

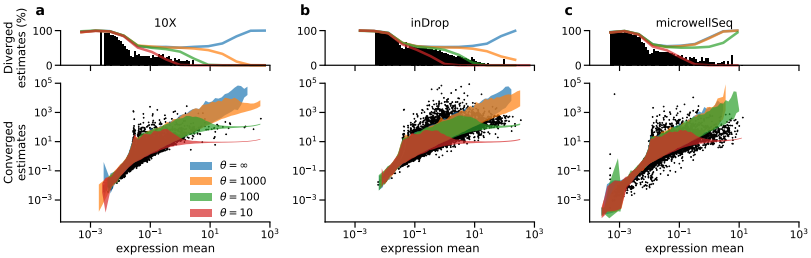

Figure S1: Overdispersion in negative control datasets. In the lower panels, each dot corresponds to one gene g. The overdispersion parameter estimates were obtained with `theta.ml()` using 100 iterations, using the expression means predicted by our model. Diverged estimates were clipped as in Figure 4g. Colors show the 5th to 95th percentile regions of the overdispersion estimates for simulated datasets with a known θ shared across genes. For each value of $\theta \in \{10, 100, 1000, \infty\}$, we simulated 1000 datasets according to Eq. 15, using means $\mu_{cg} = p_g n_c$ with empirical n_c and a series of p_g values chosen to cover the entire data range. The upper panels show the fraction of genes for which the estimate diverged (black: original data, colors: simulated data). **a:** 10x Genomics Chromium technical negative control dataset (sample 1), consisting of RNA solution split into droplets (Svensson et al., 2017). 2 000 cells. **b:** inDrop technical negative control dataset prepared as in (a) (Klein et al., 2015). 953 cells. **c:** microwellSeq biological negative control dataset from non-differentiating embryonic stem cell culture (E14 line from strain 129P2/Ola) (Han et al., 2018). 9 994 cells. Biological negative control datasets have been shown to exhibit larger variability (i.e., lower θ) compared to technical negative control datasets (Grün et al., 2014).

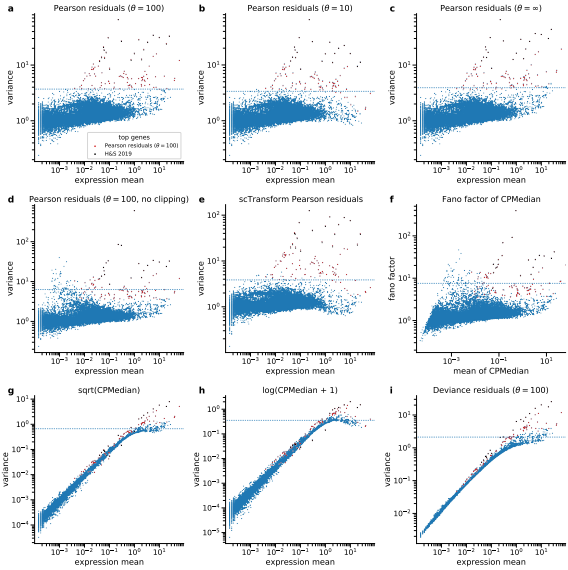

Figure S2: Comparing gene selection criteria. Each dot shows the selection criterion for a single gene in the PBMC dataset after applying a normalization method. The 100 genes with the largest criterion values in each panel lie above the dotted horizontal line. Red dots show the top 100 genes selected by the variance of Pearson residuals with $\theta = 100$. Black dots show the top 20 genes selected by `scTransform` (Hafemeister and Satija, 2019). **a:** Reproduced from Figure 5b. **b:** Analytic Pearson residuals with $\theta = 10$, roughly corresponding to `scTransform`. **c:** Analytic Pearson residuals with $\theta = \infty$, i.e., a Poisson model. **d:** Analytic Pearson residuals with $\theta = 100$ but without clipping of the residuals to $\pm\sqrt{n}$. This leads to selection of some low-expression genes. **e:** Pearson residuals of the Hafemeister and Satija (2019) model, obtained via the R implementation of `scTransform`. 87% of the top-100, and 83% of the top-1000 genes are the same as in the selection by analytic Pearson residuals with $\theta = 100$. **f:** Fano factor after sequencing depth normalization and median scaling. **g:** Variance after sequencing depth normalization, median-scaling and square-root transform. **h:** Variance after sequencing depth normalization, median-scaling and $\log(1 + x)$-transform. Note that here the variance is a non-monotonic function of the average expression. Note also that some biologically variable genes (black and red dots) in both (g) and (h) lie below the main cloud, i.e., have *lower* variance than the genes with the same average expression but without biological variability. This happens for genes that are strongly expressed but only in a small subset of cells. Square-root and log-transformations act 'too strong' on these large counts, yielding lower overall variance. **i:** Variance of deviance residuals for $\theta = 100$. Consistent with Supplementary Figure S7 in Townes et al. (2019), only high expression genes are selected.

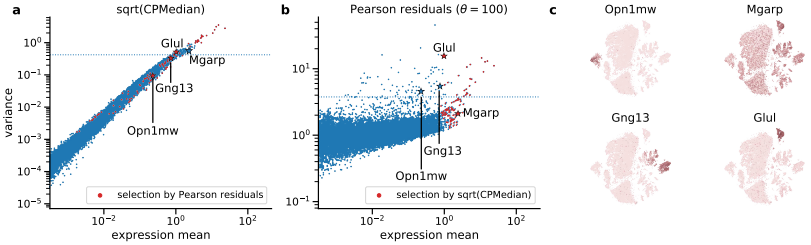

Figure S3: Selection of variable genes for a retinal dataset. All panels as in Figure 5, here showing gene selection for the largest batch of the retina dataset from (Macosko et al., 2015). It consists of replicates `p1` and `r4`–`r6`, with a total of $N = 24\,769$ cells. The cone photoreceptor marker *Opn1mv* and the bipolar cell marker *Gng13* were only selected when the variance of Pearson residuals was used as criterion. The diffusely expressed gene *Mgarp* is present in many cell types, and was only selected based on the variance of the square-root transformed data. The Müller glia marker *Glul* was selected by both methods.

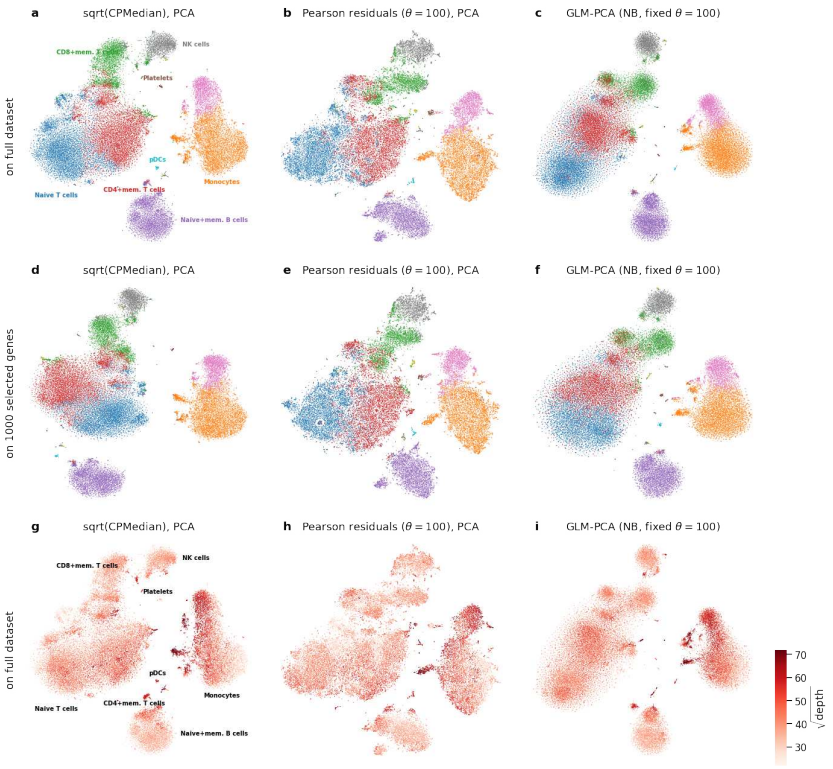

Figure S4: t-SNE embeddings based on different data transformation approaches. Each panel shows a t-SNE of the PBMC dataset, the different panels in each row show embeddings based on a different data transformation method with reduction to 50 dimensions (see Methods). Colors correspond to 10 k-means clusters provided by 10x Genomics together with the PBMC dataset. We annotated eight clusters based on known marker genes (taken from (Wagner, 2019)). **a–c:** No gene selection was used for these panels. **d–f:** Same as a–c, but using only the 1000 genes with the highest Pearson residual variance. **g–i:** Same as a–c, but overlayed with square-root-transformed sequencing depth. Values above 70 were clipped.

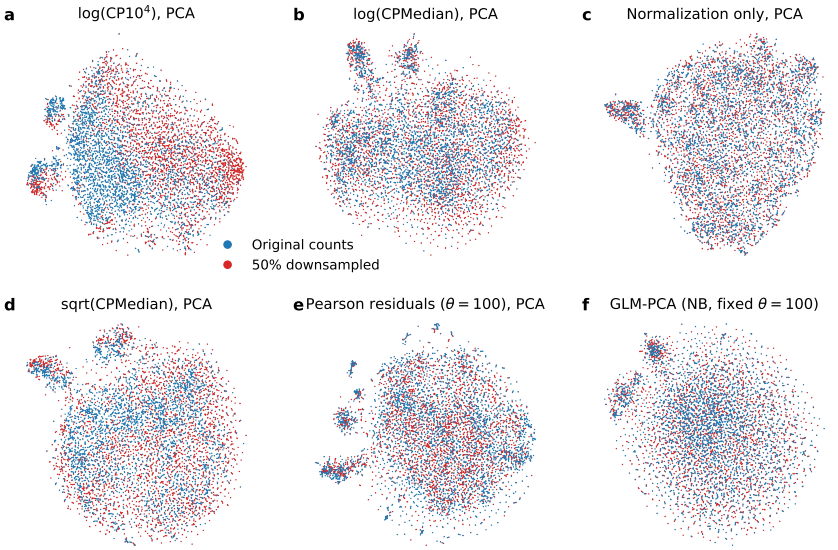

Figure S5: t-SNE embeddings of the monocyte cluster with downsampling. We analyzed the monocyte cluster from the PBMC dataset (marked orange in Figure S4, $n = 6\,307$ cells). Half of the cells (randomly selected) were downsampled to 50% of their original sequencing depth, by simulating new counts $\tilde{X}_{cg} = \text{Binomial}(n = X_{cg}, p = 0.5)$, following the analysis in Hafemeister and Satija (2019) (their Figure 6C). We then compared how well different normalization strategies remove the simulated batch-effect. PCA and t-SNE were applied after normalization as in Figure S4. **a:** Depth-normalized counts, scaled with 10^4 and then log-transformed, as in Hafemeister and Satija (2019). Note the strong batch effect due to the large scale factor. **b:** Depth-normalized counts, scaled with median depth (≈ 1500) and then log-transformed. Note that this smaller scale factor was more appropriate and removed the batch effect shown in (a). **c:** Depth-normalized counts. At the cost of not applying a variance-stabilizing transform, no batch effect remains. **d:** Depth-normalized counts, square-root transformed. A weak batch effect remains. **e:** Pearson residuals, computed with $\theta = 100$. **f:** Negative binomial GLM PCA, computed with fixed $\theta = 100$.

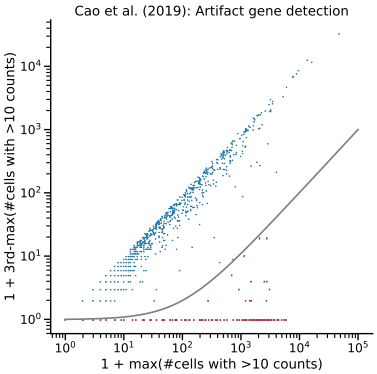

Figure S6: Artifact genes in the organogenesis dataset. Scatter plot over the 2 000 highly variable genes that we selected using Pearson residuals. For each gene and each of the 61 embryos in the data, we computed how many cells in that embryo have over 10 UMI counts of that gene. The plot compares the largest number across embryos with the third largest number. For the majority of the genes, these values were very similar, and therefore lie on the diagonal of the plot. In contrast, genes with spurious enrichment in one or two embryos lie below the diagonal. The gray line shows our exclusion criterion: 100 times difference between the largest and the third-largest number of cells with >10 UMI counts. The red dots mark 249 genes excluded for Figure 7e. Almost every embryo exhibited some of these spuriously enriched genes.

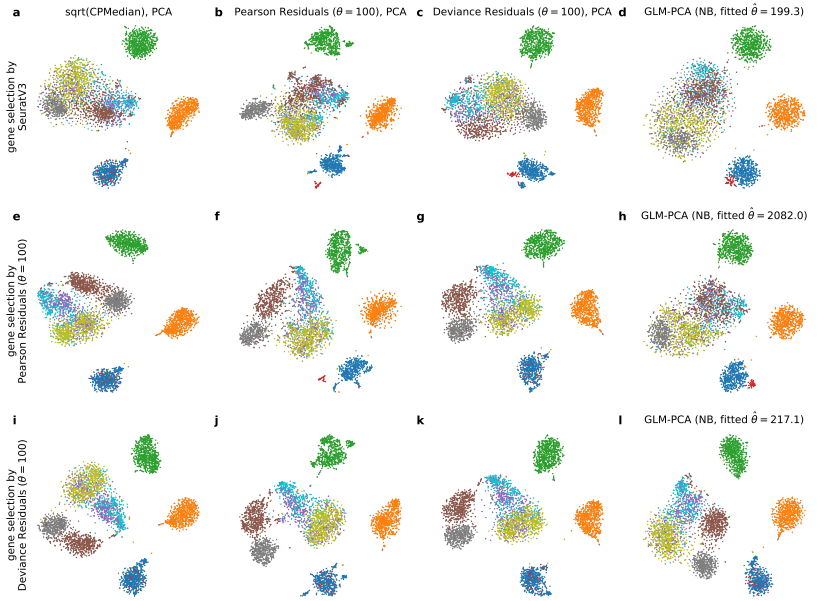

Figure S7: t-SNE embeddings of the benchmarking dataset for selected normalization pipelines. Rows show three HVG selection approaches, columns show four normalization and dimensionality reduction approaches. Each panel corresponds to one of the entries in Figure 8c.

9.2 Compound models for non-UMI counts

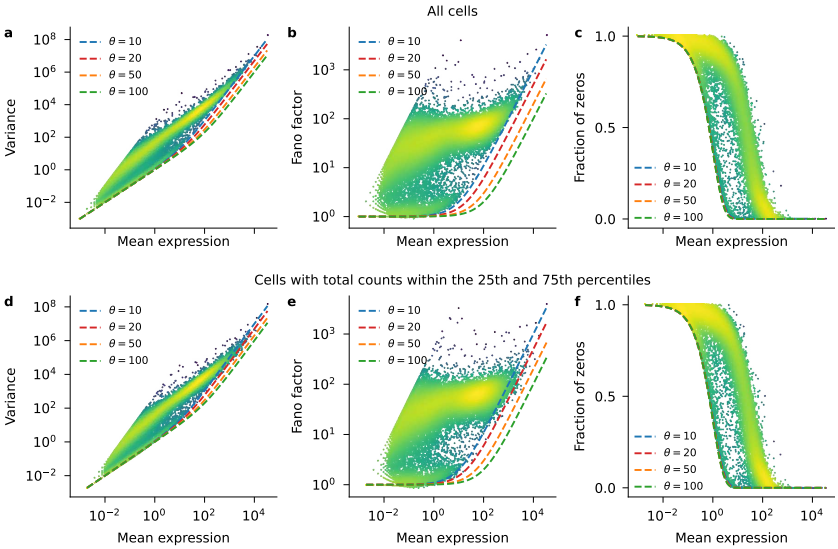

Figure S8: Sequencing depth variation increases the apparent overdispersion. Both rows are a reproduction of Figure 10, but showing pure NB models ($\alpha_Z = 1$) that differ in their overdispersion parameter θ. **a–c:** Plots based on all 1 049 cells as shown in Figure 10 with total counts per cell ranging from ∼680 000 (1^{st} percentile) to ∼2.84 million (99^{th} percentile). Note that the NB model with $\theta = 10$ fits the boundary of the data distributions. **d–f:** Plots based on a subset of the 523 cells with total counts within the 25th and 75th percentiles (from ∼1.54 million to ∼1.90 million reads). Now the NB model with $\theta = 20$ fits the boundary of the data distributions better than with $\theta = 10$. In other words, overdispersion is smaller when controlling for sequencing depth variation.

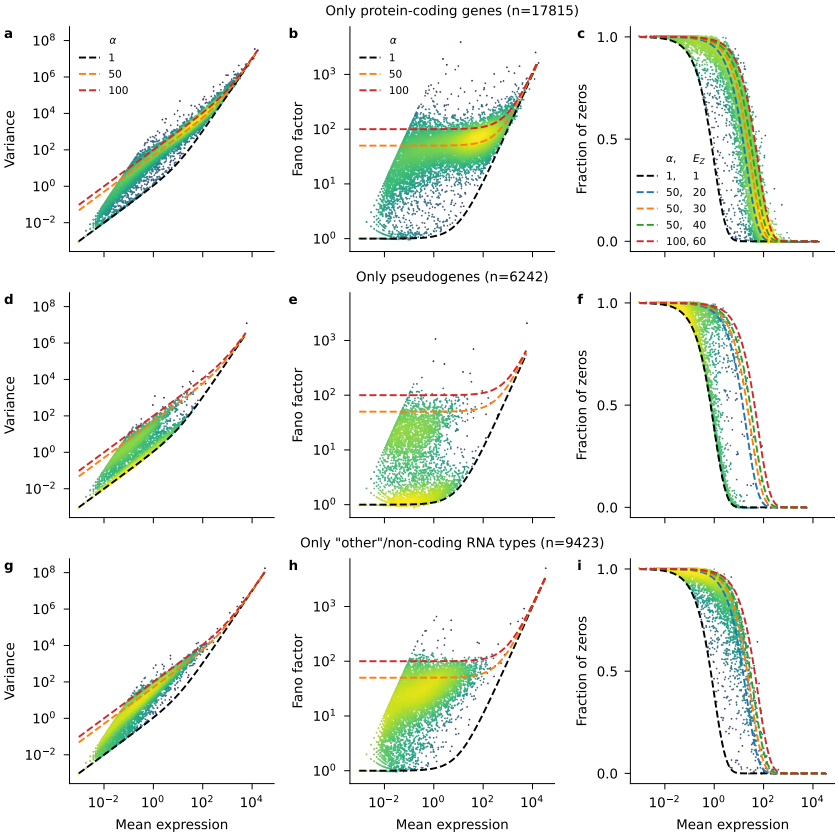

Figure S9: Non-amplified genes are mostly pseudogenes. Each row is a reproduction of Figure 10, but showing only genes from a certain category as obtained by the `mygene.info` annotation service. 434 genes without available annotation are not shown. **a–c:** Protein-coding genes. **d–f:** Pseudogenes. Many of them appear 'non-amplified' and do not follow the compound model, but rather the UMI model without amplification (black). **g–i:** Other, non-coding RNA species. Note that the bulk of these low-expression genes did not follow the compound model either.

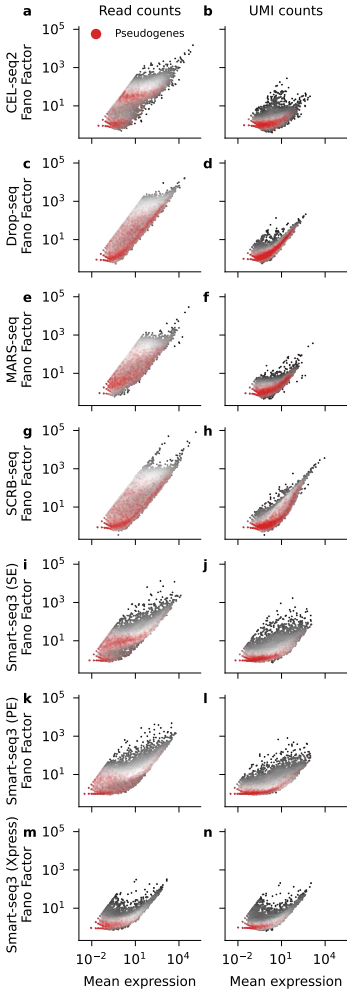

Figure S10: Pseudogenes have low Fano factor of read counts across protocols. Each panel shows the relationship between the mean and the Fano factor for all genes in each dataset. Higher density of dots is shown in brighter gray. Red dots with transparency show pseudogenes. Each row of panels shows a homogeneous dataset sequenced with a different protocol. All left-column panels are based on read counts, all right-column panels are based on UMI counts from the same dataset. Same data as in Figure 12. **a–h**: Mouse embryonic stem cells sequenced with various UMI protocols (Ziegenhain et al., 2017). For all protocols, only run A is shown. **i–l:** Smart-seq3 data from mouse fibroblasts (Hagemann-Jensen et al., 2020). SE: Single-end run. PE: Paired-end run. **m–n:** Smart-seq3 Xpress data from HEK293 cells (Hagemann-Jensen et al., 2022).

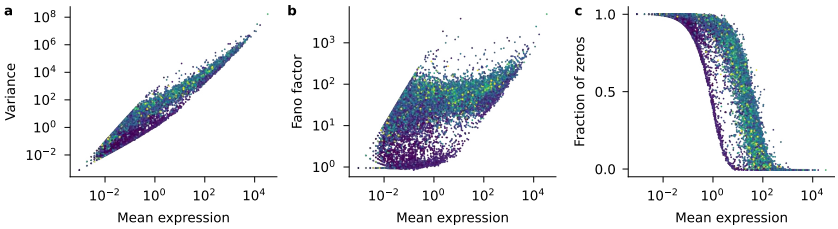

Figure S11: Pseudogenes have lower maximum transcript lengths. Reproduction of Figure 10, but showing each gene colored by the maximum length across all of its transcripts present in the `Ensembl` mouse gene database. 11 097 genes without transcript length annotation are not shown. Maximum lengths were clipped to the 98$^{\text{th}}$ percentile (9 849 bp) before plotting.

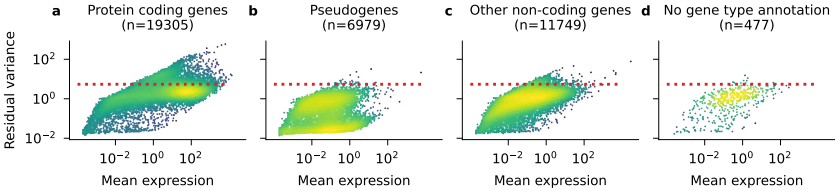

Figure S12: Genes with residual variance $\ll 1$ are mostly pseudogenes. Each panel is a reproduction of Figure 11a, showing only a certain category of genes, as in Figure S9. Each dot represents a gene and shows its mean and residual variance in the full mouse visual cortex dataset (Tasic et al., 2018). Brighter color indicates higher density of points. Red line shows cutoff for selecting 3 000 HVGs among all genes. Gene type annotations taken from the `mygene.info` service. **a:** Protein-coding genes. **b:** Pseudogenes. **c:** Other, non-coding RNA species. **d:** Genes without available annotation.

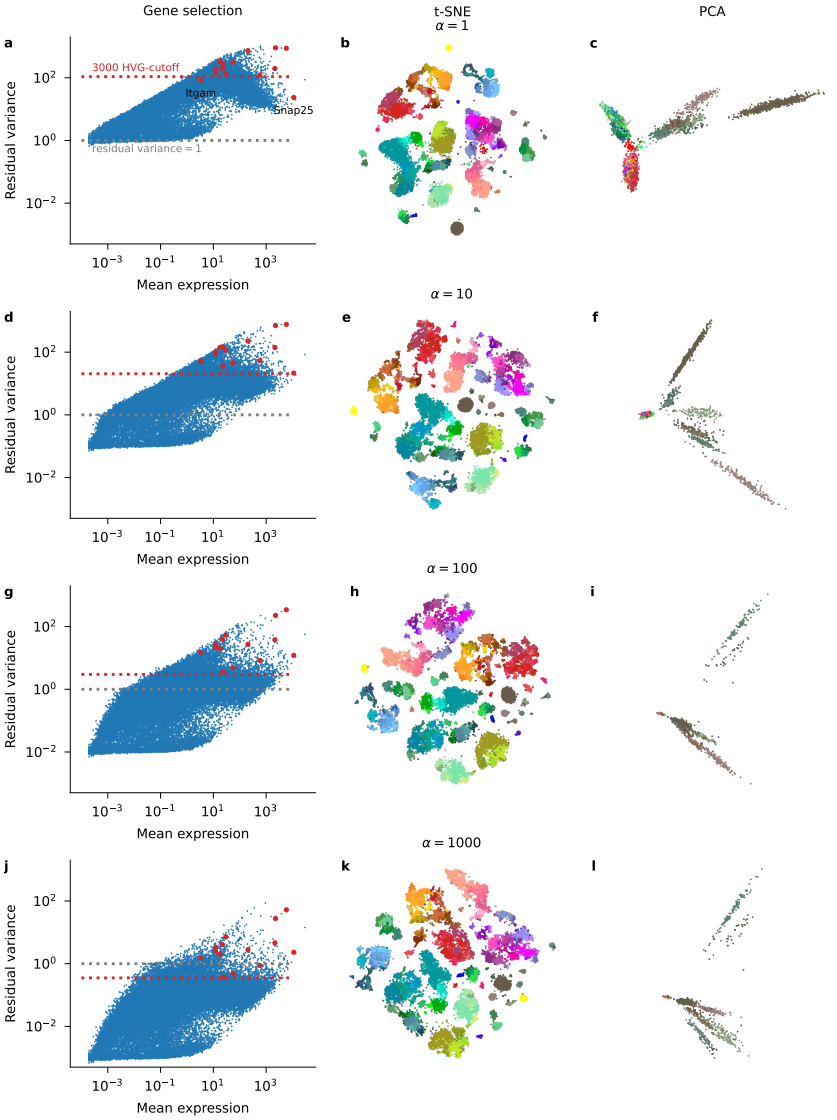

Figure S13: Influence of the amplification parameter α. Each row contains a reproduction of Figure 11a–b for various values of α_Z. The first row corresponds to the NB model without amplification, used for UMI data (Lause et al., 2021). Gray line: indicates residual variance = 1, where most non-differentially expressed genes should lie if the model is correct. All t-SNEs used the same shared initialization (see Methods). Right column shows the first two principal components (PCs) of the compound Pearson residuals.

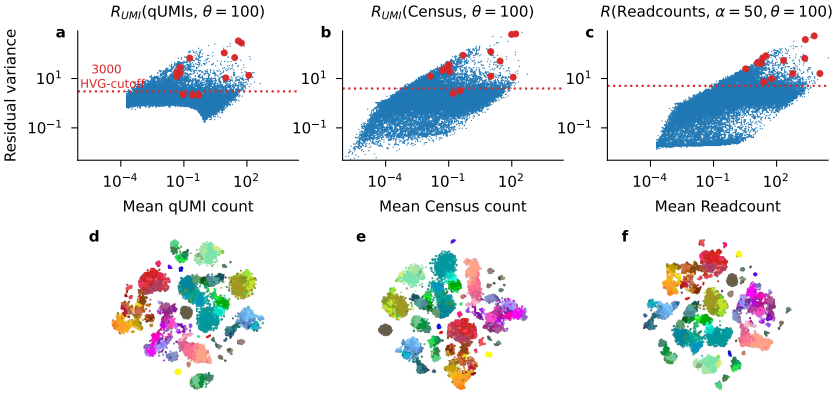

Figure S14: Comparing compound Pearson residuals to Census and qUMI. The same analysis as shown in Figure 11a–b for read counts from Tasic et al. (2018) processed with qUMIs (Townes and Irizarry, 2020) followed by UMI Pearson residuals (a, d); Census counts (Qiu et al., 2017) followed by UMI Pearson residuals (b, e); and compound Pearson residuals (our method) applied to the same set of genes (see Methods) (c, f).

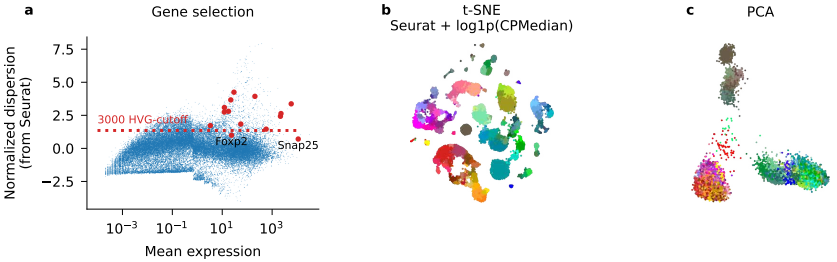

Figure S15: Preprocessing Smart-seq2 data with Scanpy default settings. The same dataset as in Figure S13, processed with `scanpy 1.9.0` defaults for normalization (counts per median normalization with `normalize_total()`, followed by `log1p()` transform) and gene selection (`flavor='seurat'`, Satija et al. (2015)) and PCA to 1 000 PCs. **a:** Seurat gene selection based on normalized dispersion. Note that two known markers (*Foxp2*, *Snap25*) are not among the top 3 000 genes selected by this method. The genes with highest normalized dispersion are markers of small, non-neural populations. **b:** t-SNE embedding of the preprocessed data. **c:** PCA embedding of the same, preprocessed data.

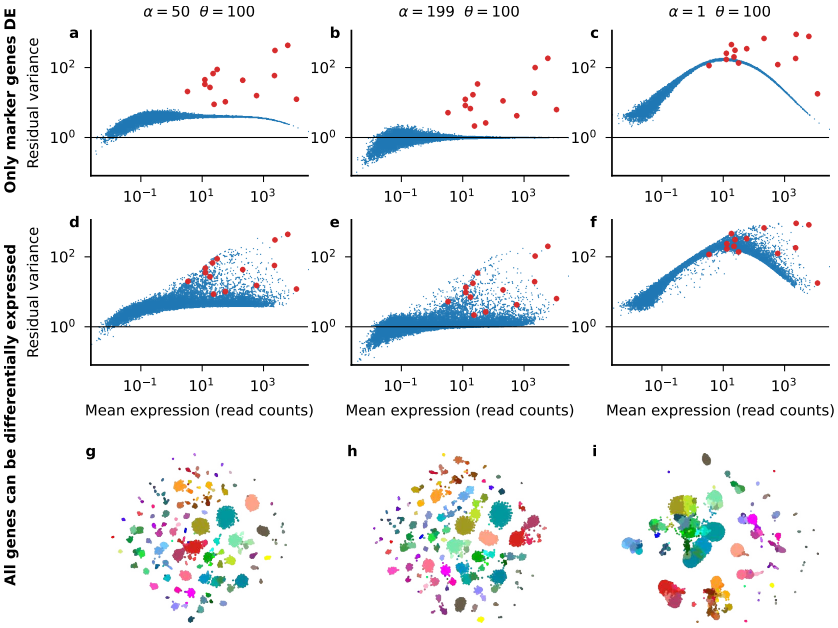

Figure S16: Compound Pearson residuals recover ground truth in realistic simulations. The same analysis as shown in Figure 11 for two simulated read count datasets that mirror the Tasic et al. (2018) cluster structure. Both simulated datasets were processed by compound Pearson residuals with three different settings: $\alpha_Z = 50$ as in Figure 11 (left); $\alpha_Z = 199$ which is the ground truth amplification factor used in this simulation (middle); $alpha_Z = 1$ corresponding to UMI Pearson residuals (right). **a–c:** Simulation I. Marker genes (red) were simulated with cluster-specific expression strengths from the Tasic et al. (2018) data, all other genes with their average expression strength across the whole dataset. Horizontal line indicates unit residual variance, expected for genes without differential expression. **d–i:** Simulation II. All genes were simulated with their cluster-specific expression strengths.

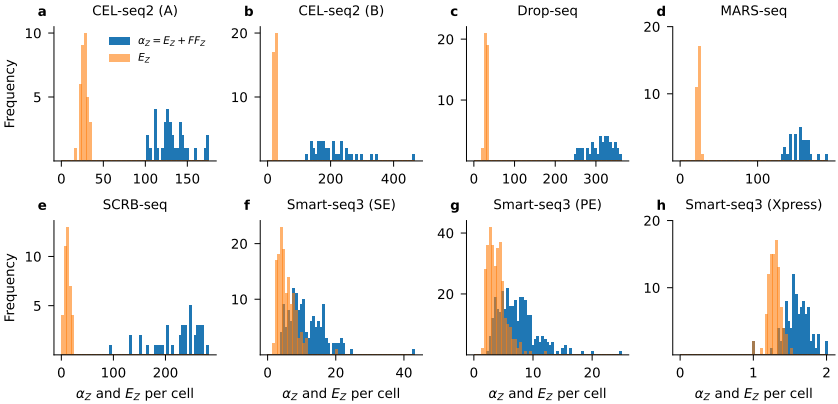

Figure S17: Cell-to-cell variability in α_Z and $\mathbb{E}[Z]$. Each panel shows amplification statistics computed per cell for all sequencing platforms listed in Table 3. Only run A is shown unless otherwise indicated.

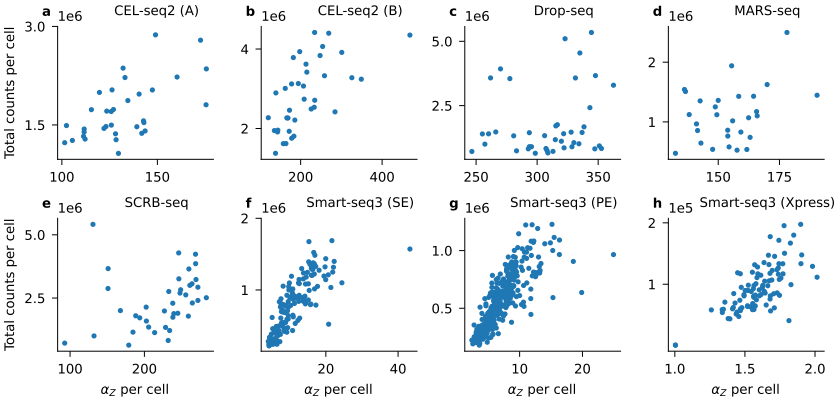

Figure S18: Total read counts are correlated with α_Z. Each panel corresponds to one of the sequencing platforms listed in Table 3; each dot is a cell. Only run A is shown unless otherwise indicated.

9.3 The art of seeing the elephant in the room

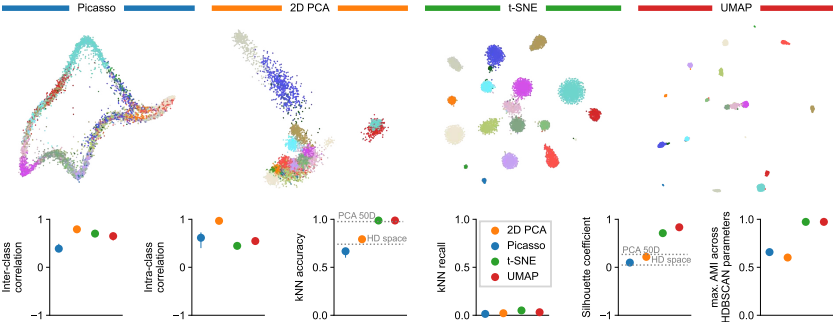

Figure S19: Simulated dataset with ground truth labels. Simulation was based on the `Ex Utero` dataset and generated 19 classes using negative binomial sampling (see Chapter Methods for details). Top row: Embeddings as in Figure 15. Bottom row: Embedding quality metrics as in Figure 16. The *k*NN recall values are very low because simulated classes do not have any internal structure. Dotted horizontal lines show the *k*NN accuracy and silhouette score in the high-dimensional gene space ("HD space") and the 50-dimensional PCA space ("PCA 50D").