Moderated designs can balance between batch-effect mitigation and cell loss due to hashtag-assisted pooling in single-cell experiments

Budha Chatterjee, Katrina Gorga, Carly Blair, et al.

Genome Res. published online July 30, 2026
Access the most recent version at doi:10.1101/gr.281624.125

Title:

Moderated designs can balance between batch-effect mitigation and cell loss due to hashtag-assisted pooling in single-cell experiments

Running title: Designs for balancing batch effect and cell loss

Authors:

Budha Chatterjee[1], Katrina Gorga[1], Carly Blair[1], Yuko Ohta[1], Michelle Radov[1], Elizabeth M. Hill[1], Christopher T. Boughter[2], Martin Meier-Schellersheim[2] & Nevil J. Singh[1 *]

Affiliation:

1. Department of Microbiology & Immunology,
   University of Maryland School of Medicine,
   685 W Baltimore St, Baltimore, MD 21201

2. Computational Biology Section,
   Laboratory of Immune System Biology,
   National Institute of Allergy and Infectious Diseases,
   National Institute of Health, Bethesda, MD 20892

Correspondence:

Budha Chatterjee, Nevil J. Singh,
Email: nsingh@som.umaryland.edu
Email: bchatterjee@som.umaryland.edu
Phone: 4107066916

Abstract:

Minimizing experimental noise is integral to robust data generation in single-cell omics. The current standard for avoiding batch effects during sample processing is barcode- or hashtag-assisted combining of different experimental treatments into one pool, allowing all samples to be subject to the technical protocols uniformly. The final datapoints for each treatment group are then computationally separated based on the original hashtag labels. Clearly, whereas hashtagging all groups and pooling them in a single well is expected to minimize batch effects, the procedure can also lead to a loss of cells that cannot be confidently decoded during the computational demultiplexing step. Here, we examine four alternate experimental designs, namely compound, reference, chain, and confounded, that could be used instead of a single-pool approach and quantify the batch effects as well as cell loss in each case. We find a linear relationship - the percentage of cells lost is double the number of hashtags used in the experiment. We use these analyses to identify experimental designs that can successfully mitigate batch effects while minimizing multiplexing, hence the cell loss, in each well. While a reference design offers the best overall performance, this study can help individual investigators choose particular approaches that are best suited for their biological questions.

Introduction:

Robust data collection, where biological effects are well preserved but technical noise is minimized, is central to the generation of rigorous data that drives genomics research. While there is an increasing reliance of the use of Single-cell RNA-seq (scRNA-seq) data to power contemporary hypothesis generation, both experiment design and data analysis in this area require extra caution. These methodologies require significant amplification of transcripts in each cell during the preparation of libraries, which can amplify well-to-well variability stemming from technical rather than biological reasons. It is widely accepted that sample multiplexing, whereby cells from different groups in the same experiment are individually barcoded and then combined in the same well, is a major innovation designed to circumvent this problem (Cheng et al. 2021; Zhang et al. 2022). Several methods for multiplexing and subsequent computational demultiplexing have been proposed (Kang et al. 2018; Guo et al. 2019; Huang et al. 2019; Shin et al. 2019; Xu et al. 2019; Srivatsan et al. 2020) with hashtag-based ones (Stoeckius et al. 2018; Gaublomme et al. 2019; McGinnis et al. 2019b; Mylka et al. 2022; Brown et al. 2023; Zhu et al. 2024) becoming more common recently. In a typical hashtag-assisted workflow, which is also used in this paper, specific samples are computationally retrieved from the sequence data, based on the expected identifying hashtags (Fig. 1A). The benefit of sample multiplexing is not just in the mitigation of batch effects (see below), but also a reduction in per-sample cost since this allows an investigator to analyze fewer cells per group in a single well, rather than dedicate the maximum capacity of each well to just one group (Pankaew et al. 2022).

'Batch effects', as we use it in this paper, is an umbrella term encompassing the variations originating from the differences in protocols, handling personnel, equipment, reagents, conditions, and duration of the experiments and subsequent sample storage, library preparation, and sequencing instruments (Tran et al. 2020; Lütge et al. 2021; Ryu et al. 2023) but not 'biological' in origin. Batch effects can be global – affecting all cell types (e.g., library prep steps) or can be local – influencing specific cell types or perhaps specific genes (e.g., storage duration/conditions affecting seq data (Courel et al. 2019)). Currently, there is no consensus on the methods of measuring batch effects in scRNA-seq data, or on how much batch variation is 'allowed' in different types of datasets when measured applying the latter disparate methods (Lütge et al. 2021; Luecken et al. 2022). Patently, considerations of batch effects are also important for meta-analysis of scRNA-seq datasets (activities commonly referred to as 'atlasing'), originating from different laboratories, following different experimental protocols, and generated through different microfluidic and sequencing technologies (Ming et al. 2022; Ryu et al. 2023; Zhang and Zhang 2024; Hrovatin et al. 2025).

To contextualize the confounding factors in each multiplexing design, it is important to accurately measure the batch effects in each case. One of the widely used metrics in the literature to measure batch

effects is entropy (Chazarra-Gil et al. 2021; Lütge et al. 2021) (in this paper 'entropy' always indicates 'normalized Shannon entropy', Fig. 1B, Methods). This metric, borrowed from information theory, is intuitively straightforward for the measurement of batch effects in single-cell data. Individual cells are projected in lower-dimensional space based on their transcriptome (or chromatin accessibility), and biologically similar cells from different 'batches' should be randomly mixed in the neighborhood of a given cell in the latter space. For each cell, its entropy is the measurement of the numeric mixing of its neighboring cells from different 'batches' (see figure legends, Fig. 1B)

Hashtag multiplexing itself has the potential to introduce noise (Fig. 1C). Apart from trapping of multiple hashtagged cells or ambiently available hashtag antibodies within each sequencing reaction droplet (see figure legends, Fig. 1C) during the microfluidic emulsification process, there is a possibility of hashtags "jumping" between cells after pooling, since the hashtags are against the same antigens (Li et al. 2024; Zhu et al. 2024). All reads from some of these latter droplets (bottom circles, Fig. 1C) will be discarded during computational demultiplexing and from downstream analyses. Thus, all multiplexing processes are inherently lossy to different extents (Stoeckius et al. 2018; Shin et al. 2019). It is generally argued that such potential loss of reads/cells due to multiplexing is the nonnegotiable collateral for batch effect mitigation. However, to our knowledge, a systematic evaluation of the batch effects and costs of different experimental design strategies, including the use of hashtag-multiplexing, has not yet been performed.

In this manuscript, we evaluate batch effects inherent in scRNA-seq pipelines by designing a single compound experiment (Fig. 2A, I). From this compound experiment, we then extract four major single-cell experimental design strategies, i.e., single pool, reference, chain, and confounded (Fig. 2A, II-VI), thereby ensuring that each design is being compared without having been processed with additional variations in wet-lab execution. Using this approach, we quantify batch effects (using entropy as the metric) as well as the cost imposed due to loss of cell recovery at the stage of hashtag demultiplexing, resulting from each experiment design strategy. As a result, our analysis allows a user to concurrently evaluate the risk of batch effects as well as the cost inflicted by losing cell recovery from any particular choice of experiment design and suggests considerations that balance both criteria.

Results

An integrated single-cell experiment for identifying batch effects across different designs

The experimental data used in this analysis are derived from an ongoing analysis of mouse splenocytes treated *in vivo* with five different variations of plasmid-encoded cytokines of the interleukin-12 family, compared to a control plasmid. Since the biology of these cytokine groups themselves is not relevant for our current analyses, we refer to this as a pool of six different samples: α, β, γ, δ, ε, ζ in

different combinations in six different wells: A, B, C, D, E, F (Fig. 2A, I). Sample α is the baseline or control group, and the rest of the samples are experimental groups. All pools received sample α. Pools A to E received one more sample other than α but no other sample. Pool F received all samples. Each sample was a 50:50 mix of female and male mice, and each received unique hashtags before pooling.

After sequencing, alignment, quality checks, and demultiplexing (Fig. 1A, Methods), we extracted six designs from the experiment for further numeric assessments. The whole experiment, which we called a 'compound' design, is denoted as design I. We then selected all α samples from all wells, and denoted this as design II (Fig. 2A, II). Next, we removed the α samples from design I – to form a 'reference' design of single-cell experiments, which is denoted as design III (Fig. 2A, III) (Song et al. 2020). We define a 'reference design' as one in which one pool contains all samples to serve as a 'reference' and the rest of the pools have only some of the samples. In design IV, we removed pool F – hence this became the 'chain' design – where pools share certain samples (sample α in this case), but there is no 'reference' pool (Fig. 2A, IV). We kept only pool F with all samples as design V (Fig. 2A, V). Lastly, in design VI, we removed all of sample α from wells A-E and samples β-ζ from well F (Fig. 2A, VI). This design is also called the 'confounded' design, where no pool shares samples and there is no reference pool. Intuitively, one might expect design I to have the least technical variation between the samples and design VI to be the noisiest.

We then processed the data in each design through three computational steps: transformation, integration across pools and measurement of batch effects (Fig. 2B). For the analysis of scRNA-seq datasets, currently there are 22 different methods for transformation (Ahlmann-Eltze and Huber 2023), 21 unique methods for integration (Tran et al. 2020; Chazarra-Gil et al. 2021; Luecken et al. 2022) (can be applied with or without highly variable gene [HVG] selection and scaling as the preprocessing step) and 8 different ways of evaluating batch effects (Lütge et al. 2021). Although there are several tool benchmarks available for these different steps (Sonrel et al. 2023), evaluation of the best pipeline, which is a combination of three steps, would be computationally prohibitive (22 transformations × 4 preprocessing options × 21 integrations × 6 designs × 8 batch effect calculations). Instead, we identified single-cell benchmarks based on the literature and used consensus tools for these different steps (tools that performed well across different benchmarks, across diverse datasets and conditions). For transformation, we chose the shifted log transform and SC-transform (Hafemeister and Satija 2019; Ahlmann-Eltze and Huber 2023). For integration methods, benchmarks are unclear (Tran et al. 2020; Chazarra-Gil et al. 2021; Luecken et al. 2022), but we use harmony (Korsunsky et al. 2019), CCA (Stuart et al. 2019) and RPCA (Hao et al. 2024) 'anchor' integrations. Of course, there are other tools with similar potential performance capabilities that are not considered here in the interest of time and effort

(Haghverdi et al. 2018; Lin et al. 2019; Song et al. 2020) (Supplemental Fig. S1, S2A-B, Methods). We used a shifted-log-transformed ('log-transformed' from here onwards) but unintegrated version for all designs (Fig. 2B, light gray arrow on the left, see Methods). Design V was not included in the integration pipelines (Fig. 2B, dark gray arrows on the right) since this involves just one pool (i.e., did not require an integration step, see Methods). The numeric projections for qualitative visualization (i.e., UMAPs) of each design, going through each combination of integration pipeline, have been documented (Supplemental Fig. S1, S2) – we show one of the 31 scenarios (2 transformations × 3 integrations × 5 designs + design V) here - the SC-transformed, RPCA-integrated design I projected as a UMAP, consisting of 33 cell clusters (Fig. 2C).

Evaluation of batch effects across experimental designs and data-analytical pipelines

Conventionally, different pools/wells are called batches (i.e., the RNA-seq library in each case is prepared separately in different tubes or wells). However, the pooled samples may also have batch effects apart from their expected biological-effect-driven variations due to possible differences in sample preparation steps. We intended to quantify the influence of integration methods on these different types of batch effects (Stuart et al. 2019; Ryu et al. 2023; Wang et al. 2024) (see Methods for the details of integration pipelines and batch-effect measurement). We, therefore, measured both types of batch effects (i.e., pool and sample) in all designs and compared the entropy distributions of integrated and unintegrated outputs for the designs. These were then compared to the cell-type entropies, as a control group, since the latter are biologically expected to remain low in contrast with those of the pools and samples (Fig. 1B, Methods).

The design with the highest median pool entropy, i.e., least batch effect (for pool), before integration was design II (median entropy = 0.842, that of design I = 0.772, design IV = 0.767, Fig. 3A, top). In this context, higher entropy indicates better mixing of cells from different pools in the lower-dimensional space. The least pool 'batch' effect in design II is consistent with previous analysis of nuclear hashing experiments reporting that hashed nuclei representing the same sample generally show negligible differences when they are processed in different wells (Gaublomme et al. 2019). We further validated this by evaluating the differentially expressed transcripts between sample α from pool A vs the rest of the pools B-F (Fig. 3B, two plots on the left, Supplemental Fig. S3A). We observed very minor (1 disparate upregulated DEG in three comparisons, 11 and 13 DEGs in the rest of the two comparisons, respectively), mostly nonoverlapping (3 common genes in comparisons with >1 DEGs) gene expression differences across five DEG comparisons, and none were significantly enriched for any biological function (Fisher's test, cutoff *P*-value = 0.01). As a positive control, we compared the samples α vs β, in pool A and in pool F from designs IV and V, respectively (Fig. 3B, plot on the right, Supplemental Fig. S3B). For these between-sample comparisons, in contrast with the within-sample comparisons (Supplemental Fig. S3A),

we observed that 55 and 97 genes were upregulated (Fig. 3B, plot on the right, Supplemental Fig. S3B), respectively, among which 47 were common. We used the median of the pool entropy for design II as a threshold (dashed line, Fig. 3A, top panel) and checked the percentage of cells in other designs that are above it. Although approximately 25-30% of cells in designs I, III, and IV were above the threshold, design VI (the confounded design) elicited a mere 11% of cells above the latter.

In the sample entropy calculations, the unintegrated design V showed the highest performance (median entropy = 0.839, that of design I = 0.791, design IV = 0.771, Fig. 3A, middle panel), as expected, since this is a reference pool where all the samples are processed together in the same well. We therefore used the median of the sample entropy distribution of design V as the sample entropy threshold (dashed line, Fig. 3A, middle panel) to check the sample 'batch' effects in the other designs. The rationale here is that any pipeline of transformation/design/integration that would yield 50% or more cells above the latter entropy threshold is performing as well as the 'biological' sample mixing observed in the reference pool (i.e., pool F - design V). Recall that pool F contains all samples of our experiment processed by identical reagents/instruments/experimentalist – hence we expect minimal sample 'batch' variation and no pool 'batch' variation at all (i.e., cells are from just one well - F). Hence, the median entropy of cells in this pool can be a useful global threshold for the acceptable extent of sample 'batch' correction in similar biological samples. Using this criterion, we find that in designs I, III, and IV, 26-32% of the cells were above the entropy threshold, supporting the consensus that semi-independent processing of the pooled samples in different wells may compound the limited sample 'batch' variations. Again, design VI was the worst performer, with only 12% of the cells above the threshold. The cell-cluster entropy distributions showed much lower medians (range of medians of entropy for designs I-VI: 0.077-0.191, Fig. 3A, bottom panel), as we would expect.

Calculating the entropy distributions post-integration (Fig. 3C, Supplemental Fig. S4) using a representative transformation/integration approach (SC-transform, RPCA integration) we found that designs I, III, and IV consistently reached ~50% of cells above the respective thresholds of pool and sample entropy distributions, consistent with the expected reduction in batch effects due to integration. However, the integration process could not recover the batch effects of design VI for either sample or pool. The performances of all transformation/integration scenarios are summarized in Fig. 3D. All integration methods in both transformations brought the desired 50% cells (dashed line, Fig. 3D) above the entropy threshold (since the thresholds were the medians from the respective contexts), except for the log-transformed CCA. For harmony and RPCA, the SC-transformation did not provide added benefits over the log transformation. Notably, it has been shown before in a previous benchmark that the latter performs "surprisingly well" (Ahlmann-Eltze and Huber 2023). To conclude, no integration method could fully recover the batch effects in design VI – i.e., designs III and IV always performed better than VI

irrespective of the subsequent computational pipeline, emphasizing that some designs are always better than others.

Efficacy of the hashtag-assisted multiplexing and demultiplexing processes

We then evaluated the labeling efficacies of the hashtags used. Droplet doublets (Fig. 1C, bottom left circle) often tend to have higher counts and features than the singlets (Luecken and Theis 2019; Germain et al. 2022). We used the HTOdemux function of the 'Seurat' package to perform the demultiplexing (Howitt et al. 2023; Li et al. 2024; Sayed et al. 2025) and found that the distributions of the doublets are largely overlapping with those of singlets, i.e., ~50% counts or features in doublets are below the 3rd quartile threshold in singlets in the distributions of counts and features from the pools A-F (Supplemental Fig. S5A, B). This suggests that a large percentage of hashtag doublets (Fig. 1C, bottom, middle, and right circles) are indeed droplet singlets (Li et al. 2024; Zhu et al. 2024). A fraction of these could also arise from droplet doublets, resulting in hashtag doublets, as reported originally (Stoeckius et al. 2018).

We reasoned that a quick way to measure the global efficacy of hashtags in a pool would be to calculate the percentage of total number of singlets against the total number of cells, within that pool (Fig. 4A, global demultiplexing efficacy, GDE). The closer this metric is to 100%, the more successful the entire pooling and demultiplexing would be. For scenarios using 2 and 4 hashtags (pools A to E), the GDE was 93% and 90% respectively. But for pool F, it falls abruptly to 76%, indicating that with a higher number of hashtags more cells are likely lost as non-singlets. To understand the relationship between use of higher number of hashtags and the number of cells lost as multiplets in a given pool, we calculated individual hashtag efficacy (IHE) which is the percentage of singlets for a given hashtag against the total number of cells that are positive for that hashtag (Fig. 4B). Several demultiplexing software, including HTOdemux, declare a lower threshold (i.e., a background level) of reads (of the specific hashtag oligomer sequence) to designate a cell to be 'positive' for that hashtag (Xin et al. 2020; Howitt et al. 2023; Klein 2023). Thus, a cell can be 'positive' for more than one hashtag (i.e., a doublet/multiplet) - ergo, singlets are the cells that are 'positive' for only one hashtag. In other words, for an individual hashtag the singlets are a subset of the positives – accordingly, IHE is the ratio of the former to the latter. IHE for the hashtags in pools A-E showed a median of ~80%. Again, in pool F, there was a sharp fall of the IHE, although the same hashtags as used in A-E were used in F, but all of them together. This means that the efficacy of the hashtags was not only dependent on the hashtags themselves but also on how many of them are being used simultaneously.

A theoretical framework for analyzing the effects of hashtag multiplexing

Intrigued by the sharp fall of GDE and IHE in pool F, we built a theoretical framework to better understand the origins of hashtag noise. We consider all cells positive for a given hashtag to be equivalent to a mathematical set. Since hashtag signals interfere with each other, potentially through different

mechanisms (Fig. 1C), these sets of positive signals can potentially overlap with each other (Fig. 4C, illustrative Venn diagrams). As the number of such sets (i.e., number of hashtags) increases, the area of overlap is expected to increase, with a parallel decrease in the exclusive parts of each set, unless all higher-order intersections (common overlap regions of multiple sets) tend to accumulate on the same cells. We derived a closed-form expression to calculate the exclusive regions (E) of a set in a universe of multiple interacting sets (Fig. 4D, gray box, Supplemental Text S1) based on the simplifying assumption that the intersections are symmetric, i.e., all hashtags show similar overlaps with each other. This closed-form expression depends on three variables: pairwise overlap x, higher-order interaction parameter y, and number of sets (i.e., hashtags) n (Methods, Supplemental Text S1). Note that $y \geq 1$, and higher values of y mean that the positive signals for multiple hashtags are increasingly not showing up in the same cells. Ergo, higher values of y will eliminate exclusive signals - as indicated by our closed-form expression. Calculating E with realistic values (see below) of x, y and n (Fig. 4E) with increasing x, y, and n, confirms that the exclusive signal decreased rapidly. Thus our theoretical framework, which is a minimalistic representation of the hashtag multiplexing and demultiplexing process, indicate that demultiplexing efficacy will increase, i.e., the abundance of singlets will increase with decrease in the number of hashtags in an experiment (lower n), with decrease in the tendency of the hashtags to participate in binary overlaps, i.e., to generate doublets (lower x) and somewhat counterintuitively, with increase in their collective propensity to generate multiplets (higher $1/y$).

Next, we estimated x and y of the hashtags in our pools (Fig. 4F, G) by randomly selecting hashtags and calculating x as the percentage of cells that are positive for both - specifically the given hashtag and the rest of the hashtags in the pool. Unfortunately, our closed-form expression cannot be explicitly solved for y when $n > 3$. Since our n is 4 or higher, we used numeric (bisection) methods to estimate y from known x, n and E. The estimates of y were always lower in pools with $n = 4$ (Fig. 4F) when compared to the estimates of y in pool F (Fig. 4G). Our model fitting confirms the intuitive understanding that with increasing the number of hashtags in a pool, the fraction of singlets keeps decreasing because of the non-overlapping lower-order (binary) intersections, i.e., rapidly diminishing higher-order intersections.

A statistical model of hashtag multiplexing

To compare the results obtained with our data to previously reported external datasets, we used a published (Stoeckius et al. 2018) dataset. We fitted the number of hashtags used in a pool to the %gap between GDE and 100 in the respective pool using nonlinear regression models first, since visual inspection of the data indicated a minor tendency towards a saturation effect (dots, Fig. 5A). The exponential decay model (Fig. 5A, R^2= 0.948) performed better than a logarithmic model fitting (R^2= 0.882). Next, for ease of interpretation, we attempted a linear regression on the same data (Fig. 5B).

Indeed, we observed a highly significant positive slope in this analysis. The quality of the fit (i.e., the R^2) was similar to the nonlinear fit. This observation again directly supported the interpretation from our theoretical framework – with an increasing number of hashtags (n), the demultiplexing efficacy decreases. The linear regression analysis further indicated that for every hashtag added to a pool, the percentage of cells lost will double (e.g., 14% cell loss for 7 hashtags). Note that currently, it is possible to super-load wells with 24 hashtags (antibody-based).

Discussion

Hashtag contamination and capturing of cells in droplets during emulsification are stochastic molecular processes. Hence, the overlaps of such interactions are expected to get compartmentalized, resulting in the loss of exclusive signals with an increase in the number of samples. Hence, the increased estimates of the parameter y in pool F (Fig. 4G), i.e., the parameter whose inverse represents the higher order interaction between multiple hashtag sets, are perhaps expected from a chemical kinetic perspective, too.

Here we carried out a single-cell experiment and generated six different data designs for the evaluation of post-integration batch-effect-mitigation, reflecting different experimental designs, and for quantifying the extent of cell loss due to hashtag-assisted super-loading. We carefully controlled well-known factors contributing to batch effects: reagents, instruments, same-day execution, and, of course, the same protocol for sample processing, library preparation, and sequencing. While our manuscript is based on a CITE-seq experiment, we speculate that our conclusions should be generally applicable to any other droplet-based single-cell sequencing technique that involves sample multiplexing through hash-tagging (Gaublomme et al. 2019; McGinnis et al. 2019b; Boughter et al. 2025).

The version of the ‘chain’ design we used here is one possible version of the chain design, which we call a ‘baseline’ chain design. There can be another ‘diagonal’ chain design which would involve α, β, γ in pool A; β, γ, ε in pool B; γ, ε, ζ in pool C, and so on. Originally, the latter was discussed in the theoretical treatment of batch correction methods of datasets with disparate cell types by Song et al (Song et al. 2020). However, we were influenced by the study that reported more stable downstream analysis results achieved through the addition of small fractions of spiked-in baseline cells (Marquina-Sanchez et al. 2020). Hence, we followed the ‘baseline’ chain design here. An additional advantage of this approach was being able to calculate the pool entropy threshold. We anticipate that the baseline chain design will also open the possibility of using post-integration, design-aware statistical finetuning for biological signal preservation across batches (Zhang et al. 2024).

In typical hashtag-based multiplexing experiments, a small population of cells can often be unlabeled (remain hashtag negative) due to technical limitations with antibody binding or antigen expression. In our analysis, we model the cells with positive signals only (Fig. 4C-E), but it should be noted that in different experimental conditions where the proportion of hashtag-negative cells is much higher than ours or previously published hashtag benchmark datasets (e.g., GSE108313), their contribution may need to be controlled for.

We proposed here a within-experiment threshold of entropy to assist in batch correction. If the majority of the cells in the integrated data are above this threshold, there is a possibility that the dataset is over-integrated and thus leads to the removal of biological signals. We would suggest investigators take appropriate precautions to prevent such scenarios, depending on the experimental and biological context that they are investigating.

Our study has limitations. The first limitation of our study is that we evaluate only a very small number of combinations of the methods of integration and transformations here. However, most of the ones we tried worked satisfactorily, i.e., the desired percentage of cells reached or crossed predefined entropy thresholds, and hence we think that if a batch effect threshold is known, the choice of the integration method should be up to the investigator, in particular since parameters in most integration methods can be further tuned to get to the threshold. Due to cost considerations, we could not include compound design experiments with pools with intermediate ($4 < n < 12$) or very high ($12 < n < 24$) numbers of hashtags, though using an external dataset helped us partially address such gaps (Fig. 5). Notably, it is possible that there is a saturation effect in cell-loss for very high numbers of hashtags, i.e., $n > 12$ (inspect the trend of the data for $n \leq 12$, in Fig. 5A), which we have not studied here. We expect that future efforts will address this gap. However, within the range of number of hashtags that we investigate here, there is a robust linear relationship between cell loss and n (Fig. 5B). We also note that we have not tested for the scenario of combinatorial hashtag indexing here (McFarland et al. 2020; Fang et al. 2021; Hwang et al. 2021). Combinatorial hashtagging or using SNPs alongside hashtags for demultiplexing are specialized, powerful methods used much less frequently for routine single-cell experiments, in our experience. Nevertheless, it will be interesting to contextualize our theoretical framework with combinatorial hash-tagging datasets. The application of doublet detection and removal techniques can further contribute to higher demultiplexing efficacy (McGinnis et al. 2019a; Germain et al. 2022). However, the benefits of prior doublet detection in demultiplexing are not well understood. Further, ambient RNA contamination across the microfluidic droplets might interfere with the quality of data integration and subsequent clustering (Yang et al. 2020). How computational decontamination algorithms influence experimental designs and subsequently, the performance of the batch correction methods needs more attention in future work. We also note that a source of variation in our dataset stems from

differences in individual samples – for instance, the specific cytokine made in the sample α vs β. Overall, this biological effect was minimal in our analysis, as we did not observe significant global transcriptional shifts from any particular treatment (Supplemental Fig. S2, sample mixing). However, this would be a factor to consider in a more rigorous benchmarking experiment. Another limitation is that moderated experimental design, which we propose as a measure for controlling batch effects and cell loss, of course, is not applicable for the literature-based atlasing activities. However, if a lab or a consortium is planning experiments to build a new atlas through fresh experiments, the moderated designs we discussed here can be quite useful to maximize cell yield and yet gracefully integrate the data.

Here we carried out a controlled comparison of whether and how different single-cell experimental designs can impact both batch effect mitigation and cell recovery. To that end, we performed one overall experiment (Fig 2A, I) and then extracted multiple designs from the data to evaluate the efficacy of the different designs at the data analytical steps of integration and demultiplexing. We reasoned that the factors influencing independent experiments, such as antibody concentrations, pipetting errors, differences in cell adhesion to each tube, antibody binding, physical aggregation of antibodies, etc., can influence both batch effect and cell recovery through further interference with the downstream analytical steps that we are focusing on here (i.e., integration and demultiplexing). Extracting the designs from one overall experiment helped us decouple experimental variations from the analytical performance metrics. Nevertheless, the next step in this inquiry would be to combine an analysis of more “real-world” experimental designs and interrogate how biological noise from disparate experiments can further modify batch effects and cell loss.

We conclude here that reference or chain multiplexing designs can reduce the number of hashtags used, and batch effects can be satisfactorily mitigated through integration of the pools. Thus, cell loss through super-loading and batch effects across pools can be balanced through refining experimental design. Moreover, in the ‘reference’ design (design III) a threshold is possible to know for the extent of biological mixing in the samples, and different integration methods can be attempted or integration parameters can be finetuned until such level of mixing is not achieved while integrating across pools. Although a compound design (design I) can provide acceptable thresholds of batch effects for both samples and pools, which can be subsequently used for optimization of the integration, the more convenient reference designs can be picked since they perform similarly well for data integration. A confounded multiplexing design, although more convenient to carry out for serially incoming samples, will be more challenging regarding correction of its batch effects. We provide here an initial rubric to choose experimental designs that are appropriate for the demands of each experiment: e.g., for patient samples or clinical isolates higher multiplexing should be preferred whereas for investigations of rare cell types or for atlas building experiments lower multiplexing will be the goal to maximize the number of

retrieved cells – both of these extremes of biological scenarios and the intermediates thereof, however, can remain within the brackets of reference or chain designs that we discussed here.

Methods:

Data generation for use in analysis

The biological experiment involved the injection of expression constructs of the IL-12 protein family as plasmids in mice using hydrodynamic injection. Different versions of plasmids expressing IL-12P40 or variations therein were used (Abdi et al. 2014; Abdi and Singh 2015), but not discussed here, since the ensuing phenotypes are not studied here. The plasmid injections were performed in mice and typically the plasmids transform cells in the liver. We then analyzed cells isolated from the spleen (Supplemental Table S1). As such, these manipulations caused minimal changes in gene expression in splenocytes. Notably, we carefully controlled well-known factors contributing to batch effects in our experiment: reagents, instruments, same-day execution, and the same protocol for sample processing, library preparation, and sequencing. The sample preparation steps were carried out by an individual experimentalist.

Sequencing and data extraction

The pooled libraries were sequenced in the Illumina NovaSeq platform (Psomagen Inc. Rockville, MD) and the base call (BCL) files demultiplexed with mkfastq function of CellRanger (10x Genomics), to generate FASTQ files containing the reads of different pools.

Alignment, data cleanup and processing

Both transcript and surface reads from each pool (i.e., index FASTQ files) were aligned with the count function from CellRanger, in default settings. The filtered barcode matrices (the output from CellRanger) were further analyzed with Seurat (v5.3.0) toolkit in R (v4.5.0) (R Core Team 2025). Six pools (A-F) were cleaned (cells with count < 20,000, 500 < feature < 3,000, mitochondrial reads < 2.5% were kept for further analysis) and demultiplexed with the default Seurat pipeline (HTOdemux function) (Supplemental Fig. S6). We removed the TCR, BCR, ribosomal and mitochondrial genes before integration since the variability within these sequences are a confounding factor independent of the design considerations of this manuscript. After cleanup we had a total of 44,846 cells from six pools (i.e., design I).

Integration of the data designs

For a detailed description of the data designs, see Figure 2A. We carried out the overall experiment (compound design) and then extracted different designs from the data for the evaluations of the quality of integration. We avoided carrying out separate experiments for each design since that would bring in additional batch effects for the disparate experiments, making the comparisons of post-integration batch effects between the designs less meaningful.

For all designs, the following steps were taken for the transformation and integration: after merging the data from different pools in the given design, first the counts were normalized with Seurat function NormalizeData. This function carries out a shifted logarithmic transformation (Ahlmann-Eltze and Huber 2023) which is referred to as 'log transformation'. Subsequently, PCA was carried out after scaling and finding 3000 most variable features, followed by cell clustering and projection (UMAP). This is unintegrated version of the designs. Next, for all designs, except design V, the following integration recipes were run at the pool level on the latter principal component space: CCA (Stuart et al. 2019), RPCA (Hao et al. 2024) and Harmony (Korsunsky et al. 2019), utilizing the IntegrateLayer function from Seurat. Further cell clustering and projecting were carried out using each of these integrated spaces. PCA was always on the first 50 dimensions, and the resolution parameter of cluster search (Louvain method) was always 1. A similar integration pipeline was carried out next, but now on the SCtransformed data (Hafemeister and Satija 2019).

The final number of clusters varied in the different designs after processing through disparate integration pipelines (Supplemental Fig. S7A). We used these numeric cell clusters to calculate the cell-type entropies (see below). As a sanity check, we manually inspected whether each cluster had orderly representation in each condition in a given design and integration pipeline (Supplemental Fig. S7B). We summarized the variation in cell abundance across clusters and conditions by averaging the coefficient of variation (CV) along with its margin of error (Supplemental Fig. S7C). Compound, reference and chain designs showed averaged CVs well below 50%. These explorations ensured that the cell-type abundances across the different conditions were largely similar, except for the context of design VI. Finally, we conducted Kruskal Wallis tests across conditions to test the null hypothesis that the cell-type abundances are sampled from the same distribution across conditions (Supplemental Fig. S7D). Encouragingly, this test did not lead to rejection of the null hypothesis in any instance.

Measurements of batch effects

We used normalized Shannon entropy (Chazarra-Gil et al. 2021; Lütge et al. 2021) for the characterization of batch effects. Briefly, Shannon entropy quantifies the information content in the neighborhood of each cell, following the expression below:

$$H(X) = -\sum_{x \in X} p(x)\, log_2 p(x) \quad (1)$$

Where X is a random variable and $p(x)$ is the probability of outcome x (Cover 1999). Let us assume, for example, that in design I with six pools, in a neighborhood of a given cell (k cells) we have a cells from pool A, c from pool C, e cells from pool E (i.e., $a + c + e = k$). Hence the 'pool entropy' of that specific cell will be:

$$H(X) = -\frac{a}{k} log \frac{a}{k} - \frac{c}{k} log \frac{c}{k} - \frac{e}{k} log \frac{e}{k} \quad (2)$$

Hence, entropy in this context is the average uncertainty of finding cells of a specific pool in the neighborhood. We used CellMixS package (Lütge et al. 2021) in R for the calculation of entropy, where a normalized Shannon entropy ($H_n(X)$) is calculated as follows:

$$H_n(X) = H(X)/\ log_2(n) \quad (3)$$

Where n is the number of categories within X. We used the evalIntegration function in CellMIxS package to this end. The normalized Shannon entropy is a unitless quantity that ranges between 0-1. The unintegrated or integrated spaces (for the different designs, integration recipes, transformations) were transferred as is to the latter function for the calculation of normalized Shannon entropies (through conversion of the Seurat objects to SingleCellExperiment objects (Amezquita et al. 2020) before using them with the evalIntegration function). Since the total number of pools, samples or cell-types (i.e., the categories, n) are very different in diverse designs, the neighborhood size, i.e., the *k*-nearest neighbors (k), should be adjusted for a fair comparison between the entropies measured for different metadata. We have set the neighborhood size (k) at three times the number of unique categories (n) for all our entropy calculations. For example, we have six pools in design I, so we set k at 18 for pool entropy calculations in design I, but to measure the cell-type entropy in design I, in Harmony integration space, we set k at 78 (n for cell clusters in Harmony space = 26). Note that $H_n(X)$ is indeterminate when $n = 1$.

As mentioned in the previous section, we carried out the integration process at the pool level since that is what we define as the traditional 'batch' in our experiment. However, we measured the batch effect at the sample level along with the pool level. The pool-level integration methods that we use here are in principle expected to correct sample-level variations also (as we observe in Fig. 3A), as long as there are shared samples, i.e., "cell states", across the pools (Stuart et al. 2019; Hao et al. 2024).

Differential expression analysis across wells

Differential expression analysis was carried out (Supplemental Fig. S3) using DESeq2 package (Love et al. 2014) in R, after pseudo-bulking the counts at the hashtag level. The differentially expressed genes were checked for enrichment of biological functions using ShinyGO (Ge et al. 2020).

Calculations of hashtag efficacy

Hashtag demultiplexing was carried out using the HTOdemux function in Seurat with parameters set to default. Thresholds declared by HTOdemux were used further to calculate the GDE and IHE (Fig. 4A, B). The latter thresholds were also utilized for enumerating the pair-wise overlap parameter, i.e., x (Fig 4C and D, Supplemental Text S1). The higher order interaction parameter, (i.e., y) was estimated using the uniroot function in R for root search within predefined boundaries. For several values of x, the estimated values of y are noted in the figure, however it can be estimated within fixed intervals for any hashtag in any pool.

Using external data

We used one external dataset (Stoeckius et al. 2018) for gathering more GDE data points in our regression analysis (Fig. 4G). This dataset (GSE108313) was made available in the convenient .rds format by the authors as part of the vignette for the Seurat HTOdemux algorithm.

Data Access

All code and data (in .h5 format) for generating the main text and supplemental figures are available in the GitHub repository: github.com/NevilLab/Hashbatch and also as Supplemental Code and Supplemental Data. All raw and processed data generated in this study have been submitted to NCBI Gene Expression Omnibus (GEO; https://www.ncbi.nlm.nih.gov/geo/) under accession number GSE333171.

Competing interest statement

None declared.

Acknowledgements

BC and NJS conceived the study. KG, CB, YO, and EMH carried out the experiment and generated the libraries. BC and CTB processed the sequencing data. BC, MR, and CTB carried out computational analyses. BC, MMS, and NJS contributed to data interpretation. NJS acquired funding. BC wrote the initial manuscript with NJS. All authors subsequently reviewed, edited, and approved the final manuscript. This project is funded by R01AI168192 from NIH and AIM-FP009 from DARPA.

References

Abdi K, Singh NJ. 2015. Making many from few: IL-12p40 as a model for the combinatorial assembly of heterodimeric cytokines. *Cytokine* **76**: 53-57.

Abdi K, Singh NJ, Spooner E, Kessler BM, Radaev S, Lantz L, Xiao TS, Matzinger P, Sun PD, Ploegh HL. 2014. Free IL-12p40 monomer Is a polyfunctional adaptor for generating novel IL-12–like heterodimers extracellularly. *The Journal of Immunology* **192**: 6028-6036.

Ahlmann-Eltze C, Huber W. 2023. Comparison of transformations for single-cell RNA-seq data. *Nature Methods* **20**: 665-672.

Amezquita RA, Lun AT, Becht E, Carey VJ, Carpp LN, Geistlinger L, Marini F, Rue-Albrecht K, Risso D, Soneson C. 2020. Orchestrating single-cell analysis with Bioconductor. *Nature methods* **17**: 137-145.

Boughter CT, Chatterjee B, Ohta Y, Gorga K, Blair C, Hill EM, Fasana Z, Adebamowo AO, Ammar F, Kosik I. 2025. CountASAP: a lightweight, easy to use python package for processing ASAPseq data. *BMC bioinformatics* **26**: 307.

Brown D, Anttila C, Ling L, Grave P, Baldwin T, Munnings R, Farchione A, Bryant V, Dunstone A, Biben C. 2023. A Risk-reward Examination of Sample Multiplexing Reagents for Single Cell RNA-Seq Genomics.

Chazarra-Gil R, van Dongen S, Kiselev VY, Hemberg M. 2021. Flexible comparison of batch correction methods for single-cell RNA-seq using BatchBench. *Nucleic acids research* **49**: e42-e42.

Cheng J, Liao J, Shao X, Lu X, Fan X. 2021. Multiplexing methods for simultaneous large-scale transcriptomic profiling of samples at single-cell resolution. *Advanced Science* **8**: 2101229.

Courel M, Clément Y, Bossevain C, Foretek D, Vidal Cruchez O, Yi Z, Bénard M, Benassy M-N, Kress M, Vindry C et al. 2019. GC content shapes mRNA storage and decay in human cells. *eLife* **8**: e49708.

Cover TM. 1999. *Elements of information theory*. John Wiley & Sons.

Fang L, Li G, Sun Z, Zhu Q, Cui H, Li Y, Zhang J, Liang W, Wei W, Hu Y. 2021. CASB: a concanavalin A-based sample barcoding strategy for single-cell sequencing. *Molecular systems biology* **17**: MSB202010060.

Gaublomme JT, Li B, McCabe C, Knecht A, Yang Y, Drokhlyansky E, Van Wittenberghe N, Waldman J, Dionne D, Nguyen L. 2019. Nuclei multiplexing with barcoded antibodies for single-nucleus genomics. *Nature communications* **10**: 2907.

Ge SX, Jung D, Yao R. 2020. ShinyGO: a graphical gene-set enrichment tool for animals and plants. *Bioinformatics* **36**: 2628-2629.

Germain P-L, Lun A, Meixide CG, Macnair W, Robinson MD. 2022. Doublet identification in single-cell sequencing data using scDblFinder. *f1000research* **10**: 979.

Guo C, Kong W, Kamimoto K, Rivera-Gonzalez GC, Yang X, Kirita Y, Morris SA. 2019. CellTag Indexing: genetic barcode-based sample multiplexing for single-cell genomics. *Genome biology* **20**: 90.

Hafemeister C, Satija R. 2019. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome biology* **20**: 296.

Haghverdi L, Lun AT, Morgan MD, Marioni JC. 2018. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology* **36**: 421-427.

Hao Y, Stuart T, Kowalski MH, Choudhary S, Hoffman P, Hartman A, Srivastava A, Molla G, Madad S, Fernandez-Granda C. 2024. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature biotechnology* **42**: 293-304.

Howitt G, Feng Y, Tobar L, Vassiliadis D, Hickey P, Dawson MA, Ranganathan S, Shanthikumar S, Neeland M, Maksimovic J. 2023. Benchmarking single-cell hashtag oligo demultiplexing methods. *NAR Genomics and Bioinformatics* **5**: lqad086.

Hrovatin K, Sikkema L, Shitov VA, Heimberg G, Shulman M, Oliver AJ, Mueller MF, Ibarra IL, Wang H, Ramírez-Suástegui C. 2025. Considerations for building and using integrated single-cell atlases. *Nature Methods* **22**: 41-57.

Huang Y, McCarthy DJ, Stegle O. 2019. Vireo: Bayesian demultiplexing of pooled single-cell RNA-seq data without genotype reference. *Genome biology* **20**: 273.

Hwang B, Lee DS, Tamaki W, Sun Y, Ogorodnikov A, Hartoularos GC, Winters A, Yeung BZ, Nazor KL, Song YS. 2021. SCITO-seq: single-cell combinatorial indexed cytometry sequencing. *Nature methods* **18**: 903-911.

Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, Wan E, Wong S, Byrnes L, Lanata CM. 2018. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature biotechnology* **36**: 89-94.

Klein H-U. 2023. demuxmix: demultiplexing oligonucleotide-barcoded single-cell RNA sequencing data with regression mixture models. *Bioinformatics* **39**: btad481.

Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, Baglaenko Y, Brenner M, Loh P-r, Raychaudhuri S. 2019. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature methods* **16**: 1289-1296.

Li L, Sun J, Fu Y, Changrob S, McGrath JJ, Wilson PC. 2024. A hybrid demultiplexing strategy that improves performance and robustness of cell hashing. *Briefings in Bioinformatics* **25**.

Lin Y, Ghazanfar S, Wang KY, Gagnon-Bartsch JA, Lo KK, Su X, Han Z-G, Ormerod JT, Speed TP, Yang P. 2019. scMerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell RNA-seq datasets. *Proceedings of the National Academy of Sciences* **116**: 9775-9784.

Love MI, Huber W, Anders S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology* **15**: 550.

Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Müller MF, Strobl DC, Zappia L, Dugas M, Colomé-Tatché M. 2022. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods* **19**: 41-50.

Luecken MD, Theis FJ. 2019. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular systems biology* **15**: e8746.

Lütge A, Zyprych-Walczak J, Kunzmann UB, Crowell HL, Calini D, Malhotra D, Soneson C, Robinson MD. 2021. CellMixS: quantifying and visualizing batch effects in single-cell RNA-seq data. *Life science alliance* **4**.

Marquina-Sanchez B, Fortelny N, Farlik M, Vieira A, Collombat P, Bock C, Kubicek S. 2020. Single-cell RNA-seq with spike-in cells enables accurate quantification of cell-specific drug effects in pancreatic islets. *Genome biology* **21**: 106.

McFarland JM, Paolella BR, Warren A, Geiger-Schuller K, Shibue T, Rothberg M, Kuksenko O, Colgan WN, Jones A, Chambers E. 2020. Multiplexed single-cell transcriptional response profiling to define cancer vulnerabilities and therapeutic mechanism of action. *Nature communications* **11**: 4296.

McGinnis CS, Murrow LM, Gartner ZJ. 2019a. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell systems* **8**: 329-337. e324.

McGinnis CS, Patterson DM, Winkler J, Conrad DN, Hein MY, Srivastava V, Hu JL, Murrow LM, Weissman JS, Werb Z. 2019b. MULTI-seq: sample multiplexing for single-cell RNA sequencing using lipid-tagged indices. *Nature methods* **16**: 619-626.

Ming J, Lin Z, Zhao J, Wan X, Ezran C, Liu S, Yang C, Wu AR, Consortium TM, Consortium T. 2022. FIRM: Flexible integration of single-cell RNA-sequencing data for large-scale multi-tissue cell atlas datasets. *Briefings in bioinformatics* **23**.

Mylka V, Matetovici I, Poovathingal S, Aerts J, Vandamme N, Seurinck R, Verstaen K, Hulselmans G, Van den Hoecke S, Scheyltjens I. 2022. Comparative analysis of antibody- and lipid-based multiplexing methods for single-cell RNA-seq. *Genome Biology* **23**: 55.

Pankaew S, Grosjean C, Quessada J, Loosveld M, Potier D, Payet-Bornet D, Nozais M. 2022. Multiplexed single-cell RNA-sequencing of mouse thymic and splenic samples. *STAR protocols* **3**: 101041.

R Core Team. 2025. R: A language and environment for statistical computing. *R Foundation for Statistical Computing, Viena, Austria*. https://www.R-project.org.

Ryu Y, Han GH, Jung E, Hwang D. 2023. Integration of single-cell RNA-seq datasets: a review of computational methods. *Molecules and cells* **46**: 106-119.

Sayed M, Wang YJ, Lim H-W. 2025. Systematic benchmark of single-cell hashtag demultiplexing approaches reveals robust performance of a clustering-based method. *Briefings in Functional Genomics* **24**: elae039.

Shin D, Lee W, Lee JH, Bang D. 2019. Multiplexed single-cell RNA-seq via transient barcoding for simultaneous expression profiling of various drug perturbations. *Science advances* **5**: eaav2249.

Song F, Chan GMA, Wei Y. 2020. Flexible experimental designs for valid single-cell RNA-sequencing experiments allowing batch effects correction. *Nature communications* **11**: 3274.

Sonrel A, Luetge A, Soneson C, Mallona I, Germain P-L, Knyazev S, Gilis J, Gerber R, Seurinck R, Paul D. 2023. Meta-analysis of (single-cell method) benchmarks reveals the need for extensibility and interoperability. *Genome Biology* **24**: 119.

Srivatsan SR, McFaline-Figueroa JL, Ramani V, Saunders L, Cao J, Packer J, Pliner HA, Jackson DL, Daza RM, Christiansen L. 2020. Massively multiplex chemical transcriptomics at single-cell resolution. *Science* **367**: 45-51.

Stoeckius M, Zheng S, Houck-Loomis B, Hao S, Yeung BZ, Mauck WM, Smibert P, Satija R. 2018. Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. *Genome biology* **19**: 1-12.

Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, Hao Y, Stoeckius M, Smibert P, Satija R. 2019. Comprehensive integration of single-cell data. *cell* **177**: 1888-1902. e1821.

Tran HTN, Ang KS, Chevrier M, Zhang X, Lee NYS, Goh M, Chen J. 2020. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome biology* **21**: 12.

Wang Y, Thistlethwaite W, Tadych A, Ruf-Zamojski F, Bernard DJ, Cappuccio A, Zaslavsky E, Chen X, Sealfon SC, Troyanskaya OG. 2024. Automated single-cell omics end-to-end framework with data-driven batch inference. *Cell Systems* **15**: 982-990. e985.

Xin H, Lian Q, Jiang Y, Luo J, Wang X, Erb C, Xu Z, Zhang X, Heidrich-O'Hare E, Yan Q. 2020. GMM-Demux: sample demultiplexing, multiplet detection, experiment planning, and novel cell-type verification in single cell sequencing. *Genome biology* **21**: 188.

Xu J, Falconer C, Nguyen Q, Crawford J, McKinnon BD, Mortlock S, Senabouth A, Andersen S, Chiu HS, Jiang L. 2019. Genotype-free demultiplexing of pooled single-cell RNA-seq. *Genome biology* **20**: 290.

Yang S, Corbett SE, Koga Y, Wang Z, Johnson WE, Yajima M, Campbell JD. 2020. Decontamination of ambient RNA in single-cell RNA-seq with DecontX. *Genome biology* **21**: 57.

Zhang Y, Xu S, Wen Z, Gao J, Li S, Weissman SM, Pan X. 2022. Sample-multiplexing approaches for single-cell sequencing. *Cellular and Molecular Life Sciences* **79**: 466.

Zhang Z, Mathew D, Lim TL, Mason K, Martinez CM, Huang S, Wherry EJ, Susztak K, Minn AJ, Ma Z. 2024. Recovery of biological signals lost in single-cell batch integration with CellANOVA. *Nature Biotechnology*: 1-17.

Zhang Z, Zhang X. 2024. Data-driven batch detection enhances single-cell omics data analysis. *Cell Systems* **15**: 893-894.

Zhu Q, Conrad DN, Gartner ZJ. 2024. deMULTIplex2: robust sample demultiplexing for scRNA-seq. *Genome Biology* **25**: 37.

Figure 1: A cartoon overview of a typical single-cell experiment, estimation of batch effects and hashtag labeling. (A) Overview of an in-house, single-day, scRNA-seq experiment in a laboratory. Samples are labeled with hashtags and combined for multiplexing. Each multiplexed pool is loaded into separate wells on a microfluidic plate (e.g., 10x Genomics chips) for droplet capture and reaction. The libraries are sequenced, and reads from each well (i.e., the pool) are separated based on the indexes they receive during library preparation. FASTQ files are then taken through the computational pipelines of alignment, quality control, demultiplexing etc. Finally, the data from all the wells in that experiment are integrated to mitigate batch effects. (B) In scRNA-seq data analysis pipelines, the cells are projected in lower-dimensional spaces (e.g., principal component space or further corrected integrated spaces) based on their transcriptome. Generally, 30-50 dimensions are used, but in the cartoon shown here, a two-dimensional space is depicted for illustration. Each cell can have different attributes, also known as metadata, as depicted by different colors here within each subplot: sample (left), pool (middle), cell-type (right), etc. In the absence of batch effects, similar cells from different pools and samples will be well-mixed in this space, since they will not have batch-related differences in their gene expression, other than biologically relevant differences. The lower the batch effects, the higher the diversity of a cell's neighborhood and the larger the entropy of that cell for that particular metadata. Typically, cell-type metadata has low mixing compared to those of samples and pools, since cells of the same type are expected to remain together in the reduction space. (C) A cartoon of emulsion droplets with hashtag-labeled cells. Generally, they are individual cells with one predominant hashtag (top circles). In some cases, more than one cell can be in a droplet (left bottom circle). Or one cell may be labeled by more than one hashtag due to ambient hashtag antibodies after pooling (middle bottom). Alternatively, the emulsification process might capture free antibodies in the medium (right bottom). *This figure is created with BioRender.com*.

Figure 2: Different designs of experimental data and succeeding analytical recipes. (A) Different designs of data that are considered for batch effect calculations. Columns in each chart indicate wells (i.e., pools A to F), rows indicate samples (α to ζ, and hashtags: 1 to 9, 12 to 14). Each design is indicated by colored boxes on each chart. The lighter blue boxes indicate four times higher loading of cells than those in navy blue. The number of cells in each design and the design numbers in Roman numerals are shown in colors matching those used in the next figures. Design I illustrates the compound format used in the entire experiment (design I). Descriptions of designs II to VI are given in the text. (B) A schematic of the analytical pipeline of taking the designs through the transformations, integrations, and batch effect calculations. Light gray block arrows: all designs; dark gray block arrows: all except design V. See text for the details of the tools used under each step. Each instance of the pipeline consists of one item from each of the three boxes in the diagram, i.e., designs, transformations and integrations. The latter is skipped

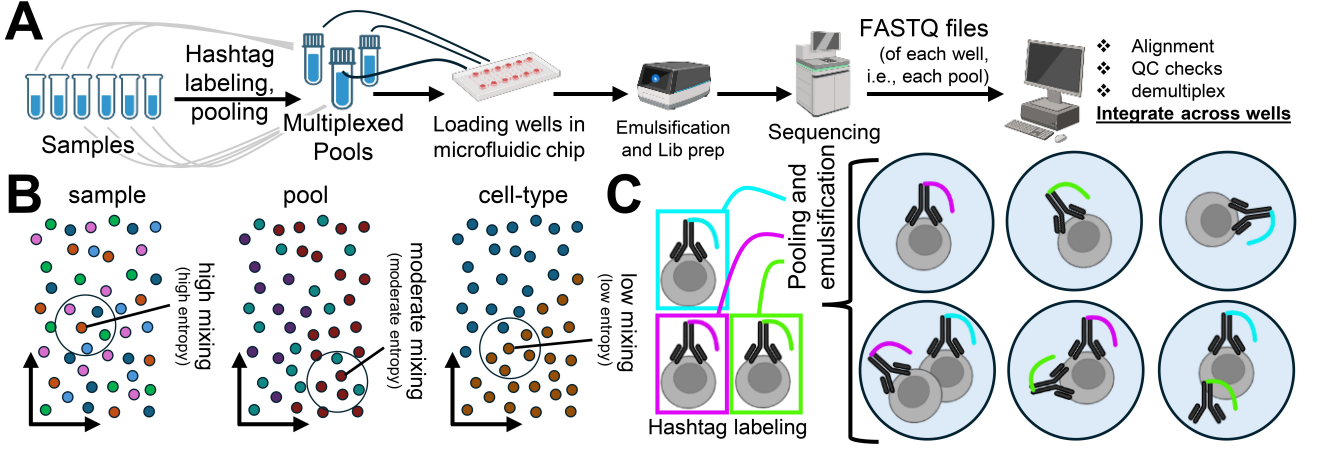
for the 'unintegrated' processing (light gray arrow) as indicated in subsequent figures. (C) UMAP on RPCA integration on the entire data, i.e., design I, post SC-transformation. Contrasting colors are used for the adjacent clusters for the ease of visualization. The keys to the colors of the cell clusters are indicated in the box at the bottom. The convention of color keys used in this specific UMAP is not followed in any other figure panel in this paper. The median representative points of each cluster are indicated with a black dot along with immediately adjacent annotations of the respective cluster IDs. The annotated medians are to be used in conjunction with the color keys at the bottom – similar colors may have been used for two clusters in a limited number of instances, such as 1 and 18, 4 and 10, etc., due to restricted availability of adequately contrasting colors. These instances of using the same color for two clusters are carefully chosen such that the specific pair of clusters is far from each other in the projection space. Clusters 26 and 27 are split in the UMAP (splitting of cell clusters due to projection is not unusual and is observed routinely). Thus, for instance, the median of cluster 27 lies in the middle of two split populations. The medians are plotted for the ease of visualization only – they are not related to the clustering algorithm used (see Methods).

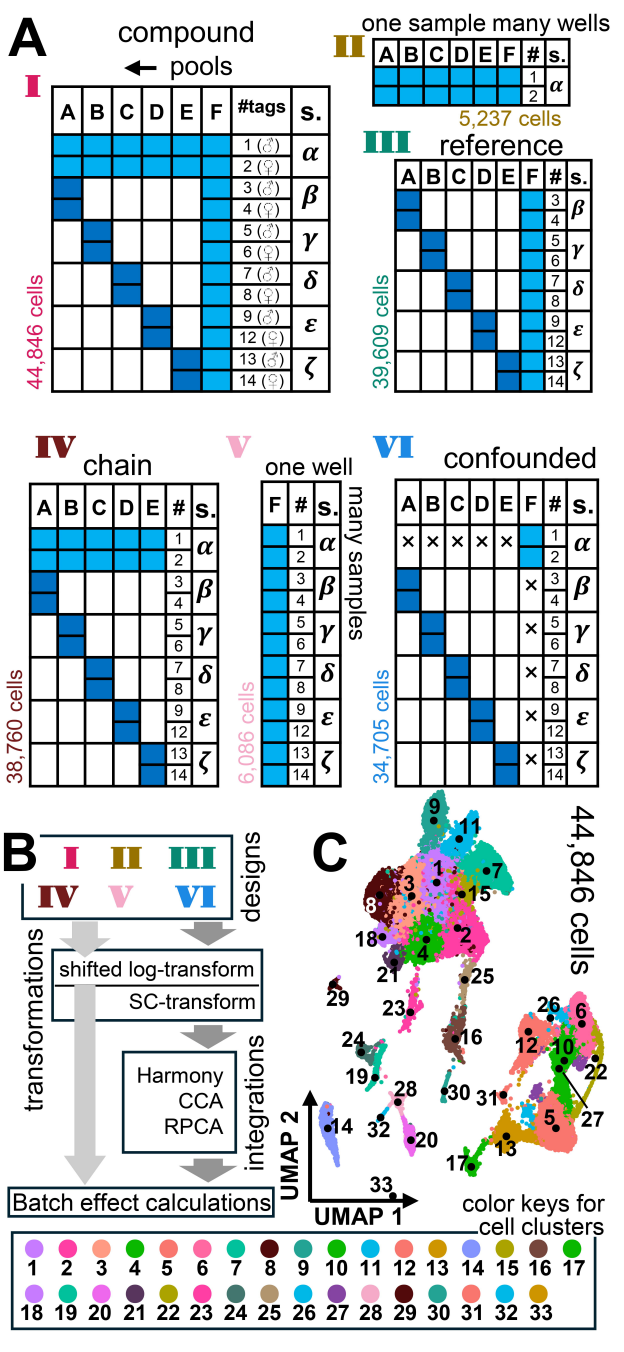
Figure 3: Estimation of batch effects in different designs pre- and post-data integration. (A) Entropies of each cell in different designs are measured, and the distributions of the entropies are presented as violin plots. Entropies of pools (top), samples (middle), and cell-clusters (i.e., cell-types) (bottom) are shown. For design V and design II, pool and sample entropies, respectively, are not shown (see Fig. 2A, text, Methods). The percentages on the X-axis are percentages of cells above the threshold (dashed gray line) that was chosen as the median of the pool entropy for design II. Unint: unintegrated; LT: log-transformed. (B) Representative volcano plots to compare gene expression in cells from (left) the same sample but different wells and (right) the same well but different samples. All plots are shown in Supplemental Figure S3. Dashed horizontal lines indicate p = 0.01. The colors in the volcano plots are for the ease of visualization and are not linked to the color schemes in any other figures. (C) Same as (A) but looking at the mixing in SC-transformed RPCA (see text) integrated reduced space. Design V is excluded in this representation since this design has only one well, hence cannot be integrated. SCT: SC-transformed. (D) Percentage of cells above the pool (left) and sample (right) entropy thresholds obtained with three different integration strategies (Harmony, CCA, RPCA; see text) for designs I, III, IV and VI. The dashed line represents 50%.

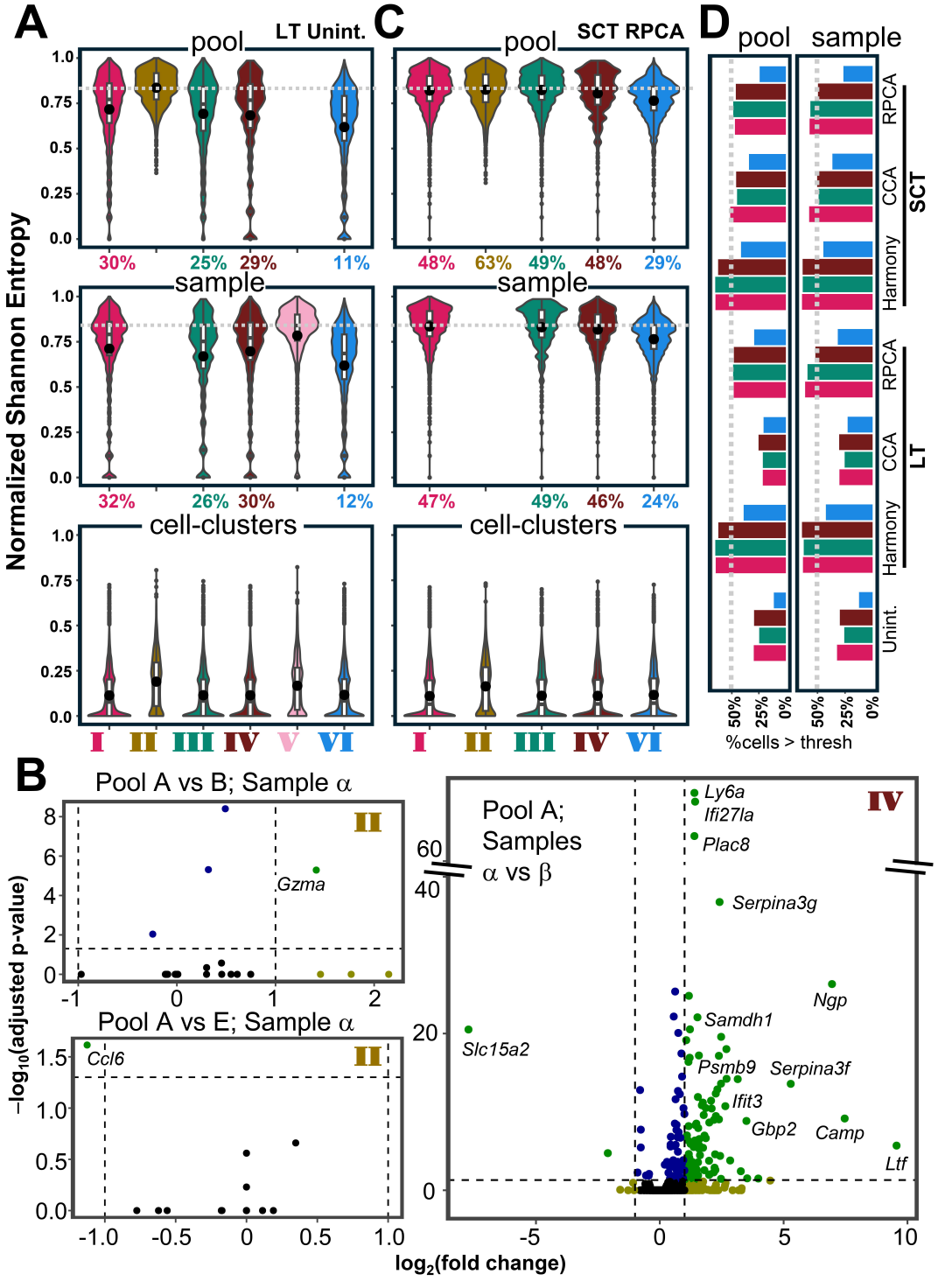
Figure 4: Evaluation of the performance of hashtags in demultiplexing: (A) Evaluation of global demultiplexing efficacy (GDE) in each pool. Bar diagram showing percentage of hashtag singlets within all cells in a well. The first five bars are the calculations of GDEs of wells A to E, combining two hashtags within a sample as one. The approximated mean GDE is noted on the boxes covering the lower parts of the bars. (B) Evaluation of individual hashtag efficacies (IHE) for each hashtag used in each pool shown as dot plots accompanied by overlaid box plots. Pools and the number of hashtags (in brackets) used in them are marked on the X-axis. The hash (#) signs are abbreviations for 'hashtag'. For (A) and (B), the expressions of GDE and IHE, respectively, are noted in the gray boxes at the top. (C) Cartoon Venn diagrams to illustrate the relations between overlaps and the availability of exclusive parts of intersecting sets. (D) A closed-form expression to calculate exclusive parts of sets (E), given that we know x, y and n (see text, Supplemental text S1) in the dark gray box. (E) Charts showing calculations of E, for the given value of y and x (shown at the top of the charts). The different values of x are at the top row,

followed by ∩ (intersection) sign, and n is the first column in the charts. (F, G) Estimation of x and y in single-cell data. Three charts are for the pools with four hashtags – the first columns indicate the specific hashtag for which the overlap is calculated and the second columns are the estimated values of x. (F). Two bar plots at the bottom are for pool F, i.e., 12 hashtags (G). The title above the charts/bar plots, such as 'F, #4', indicate that x and y are estimated for hashtag 4 in well F. Below that, the percentage of exclusive signals, i.e., singlets for that hashtag, is noted. The bars or the second columns of the charts are the values of x, as estimated directly from data (see Methods). Numeric estimates of y are noted immediately below the bars or the chart entries (i.e., the corresponding x values).

Figure 5: Statistical modeling of hashtag demultiplexing: (A) The difference between GDE and 100 (as percentages) is plotted on Y-axis for different pools, while the number of hashtags in respective pools in the X-axis, from the current dataset and another external dataset (GSE108313). Nonlinear regression analysis was carried out using the exponential decay model (the increasing form, equation at the bottom) on the dataset. The blue dashed line represents the best fit nonlinear model. R^2 of nonlinear regression is mentioned on the left top. The estimated values of the parameters a, b and c with 95% CI are as follows: $a = 32.6$ (14.53, 50.68), $b = 0.08661$ (-0.001616, 0.1748), $c = 1.3$ (-1.477, 4.077). (B) Linear regression on the same dataset as in (A). The dashed line is the best-fit linear model. Slope, *P*-value, and R^2 of linear regression on the data points are recorded on the left top.

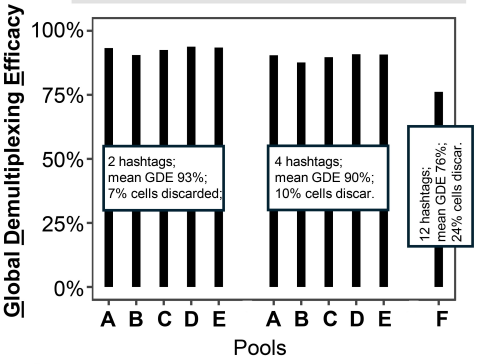

A
Samples
Hashtag labeling, pooling
Multiplexed Pools
Loading wells in microfluidic chip
Emulsification and Lib prep
Sequencing
FASTQ files
(of each well, i.e., each pool)
Alignment
QC checks
demultiplex
Integrate across wells
B
sample
pool
cell-type
high mixing
(high entropy)
moderate mixing
(moderate entropy)
low mixing
(low entropy)
C
Pooling and emulsification
Hashtag labeling


A
compound
pools
I
A B C D E F
#tags
s.
44,846 cells
α
β
γ
δ
ε
ζ
II
one sample many wells
A B C D E F
#
s.
5,237 cells
III
reference
39,609 cells
IV
chain
38,760 cells
V
one well
many samples
6,086 cells
VI
confounded
34,705 cells
B
designs
transformations
shifted log-transform
SC-transform
integrations
Harmony
CCA
RPCA
Batch effect calculations
C
44,846 cells
UMAP 2
UMAP 1
color keys for
cell clusters


A
LT Unint.
C
SCT RPCA
D
pool
sample
Normalized Shannon Entropy
1.0
0.75
0.5
0.25
0.0
30%
25%
29%
11%
48%
63%
49%
48%
29%
32%
26%
30%
12%
47%
49%
46%
24%
cell-clusters
I
II
III
IV
V
VI
RPCA
CCA
Harmony
SCT
LT
Unint.
50%
25%
0%
%cells > thresh
B
Pool A vs B; Sample α
Gzma
Pool A vs E; Sample α
Ccl6
−log10(adjusted p-value)
Pool A;
Samples
α vs β
Ly6a
Ifi27la
Plac8
Serpina3g
Ngp
Samdh1
Slc15a2
Psmb9
Serpina3f
Ifit3
Gbp2
Camp
Ltf
log2(fold change)

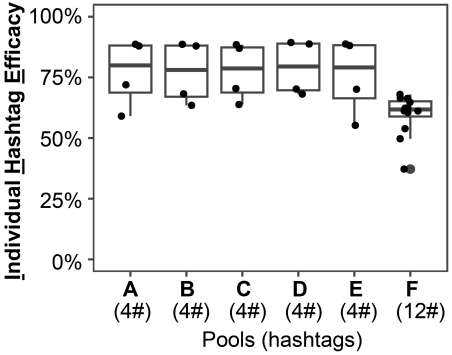
A

$$GDE\% = \frac{singlets}{singlets + multiplets + negatives}\%$$

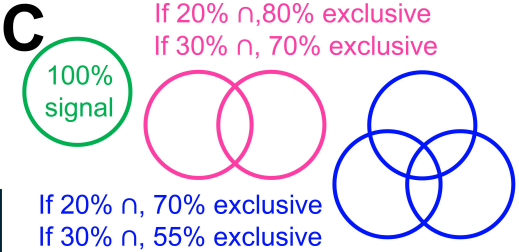


B

$$IHE\% = \frac{singlets\ for\ a\ given\ hashtag}{positives\ for\ a\ given\ hashtag}\%$$



C



D

x = pair-wise overlap;
y = higher order interaction param;
n = number of sets.

$$E = 1 - xy + x\frac{(y-1)^{n-1}}{y^{n-2}}$$

E

y = 2

n	∩ 2.5%	∩ 5%
2	97.5%	95.0%
4	96.6%	91.3%
6	95.1%	90.3%
10	95.0%	90.0%

y = 2

n	∩ 10%	∩ 15%
2	90.0%	85.0%
4	82.5%	73.8%
6	80.6%	70.9%
10	80.0%	70.1%

y = 5

n	∩ 2.5%	∩ 5%
2	97.5%	95.0%
4	93.9%	87.8%
6	91.6%	83.2%
10	89.2%	78.4%

y = 5

n	∩ 10%	∩ 15%
2	90.0%	85.0%
4	75.6%	63.4%
6	66.4%	49.6%
10	56.7%	35.1%

y = 10

n	∩ 2.5%	∩ 5%
2	97.5%	95.0%
4	93.2%	86.5%
6	89.8%	79.5%
10	84.7%	69.4%

y = 10

n	∩ 10%	∩ 15%
2	90.0%	85.0%
4	72.9%	59.4%
6	59.0%	38.6%
10	38.7%	8.1%

F

A, #4
E = 87.9%

#	x
1	2.1%
2	3.8%
3	6.5%

y = 2.2

C, #8
E = 88.5%

#	x
1	2.8%
2	3.6%
7	6.1%

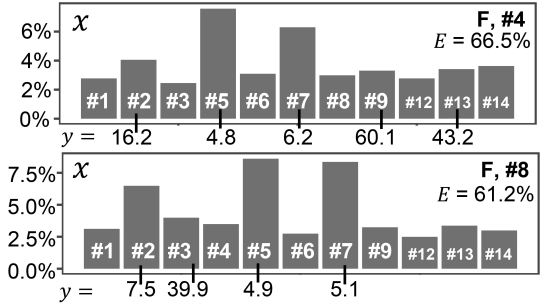
y = 2.3

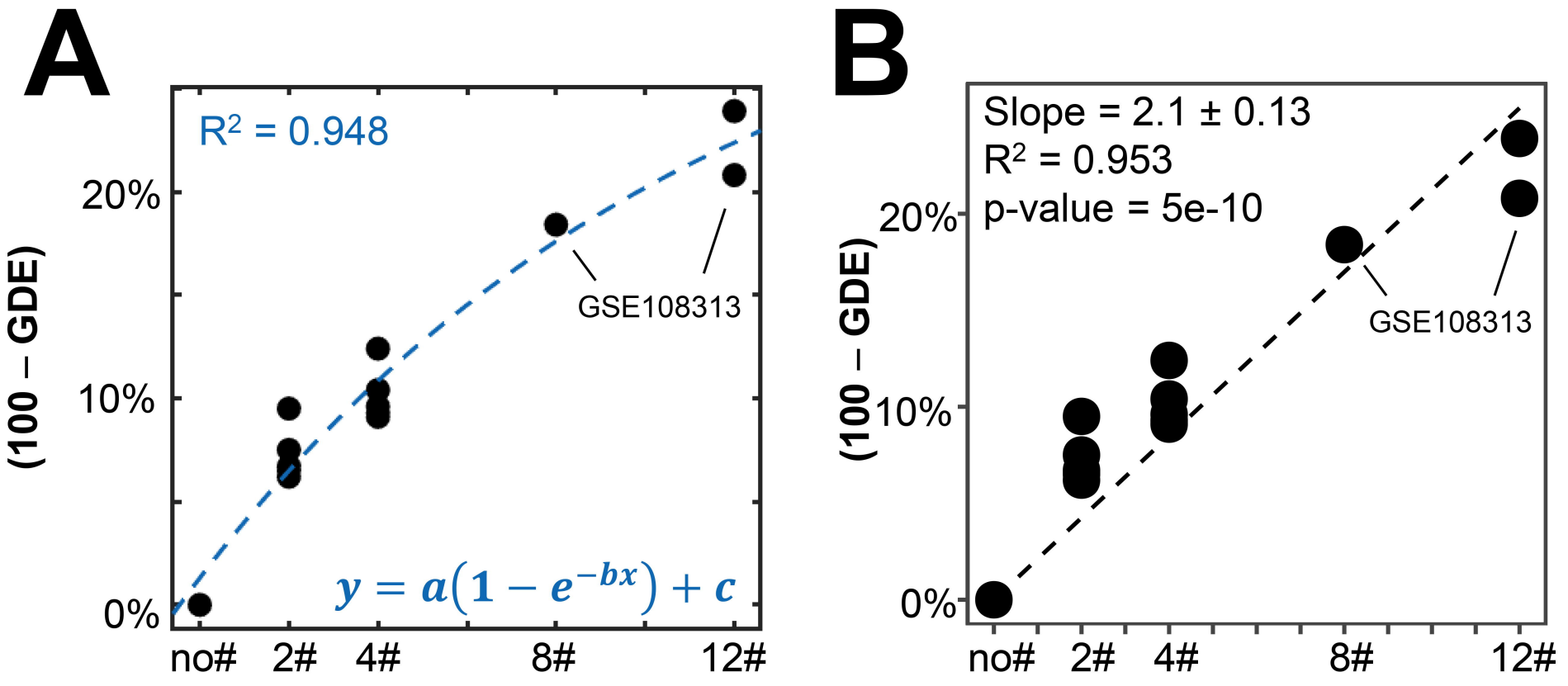
E, #14
E = 88.8%

#	x
1	2.6%
2	3.1%
13	6.1%

y = 2.2

G




A
R² = 0.948
20%
10%
0%
(100 − GDE)
GSE108313
y = a(1 − e^−bx) + c
no#
2#
4#
8#
12#
B
Slope = 2.1 ± 0.13
R² = 0.953
p-value = 5e-10
20%
10%
0%
(100 − GDE)
GSE108313
no#
2#
4#
8#
12#