HHS Public Access

Author manuscript

Nat Methods. Author manuscript; available in PMC 2021 August 15.

Published in final edited form as:

Nat Methods. 2021 March ; 18(3): 272–282. doi:10.1038/s41592-020-01050-x.

Joint probabilistic modeling of single-cell multi-omic data with totalVI

Adam Gayoso[1,*], **Zoë Steier**[2,*], **Romain Lopez**[3], **Jeffrey Regier**[4], **Kristopher L Nazor**[5], **Aaron Streets**[1,2,6,†], **Nir Yosef**[1,3,6,7,†]

[1]Center for Computational Biology, University of California, Berkeley, Berkeley, CA, USA

[2]Department of Bioengineering, University of California, Berkeley, Berkeley, CA, USA

[3]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA

[4]Department of Statistics, University of Michigan, Ann Arbor, Ann Arbor, MI, USA

[5]BioLegend, Inc., San Diego, CA, USA

[6]Chan Zuckerberg Biohub, San Francisco, CA, USA

[7]Ragon Institute of MGH, MIT and Harvard

Abstract

The paired measurement of RNA and surface proteins in single cells with CITE-seq is a promising approach to connect transcriptional variation with cell phenotypes and functions. However, combining these paired views into a unified representation of cell state is made challenging by the unique technical characteristics of each measurement. Here we present Total Variational Inference (totalVI; https://scvi-tools.org), a framework for end-to-end joint analysis of CITE-seq data that probabilistically represents the data as a composite of biological and technical factors including protein background and batch effects. To evaluate totalVI's performance, we profiled immune cells from murine spleen and lymph nodes with CITE-seq, measuring over 100 surface proteins. We demonstrate that totalVI provides a cohesive solution for common analysis tasks like dimensionality reduction, the integration of datasets with different measured proteins, estimation of correlations between molecules, and differential expression testing.

†Corresponding author: astreets@berkeley.edu, niryosef@berkeley.edu.
*These authors contributed equally.

Author contributions

A.G. and Z.S. contributed equally. A.G., Z.S., A.S., and N.Y. designed the study. A.G., Z.S, R.L., J.R., and N.Y. conceived of the statistical model. A.G. implemented the totalVI software with input from R.L. K.L.N. designed and produced antibody panels and provided input on the study. Z.S. designed and led experiments with input from A.S. and N.Y. A.G. and Z.S. designed and implemented analysis methods and applied the software to analyze the data with input from A.S. and N.Y. A.S. and N.Y. supervised the work. A.G., Z.S., R.L., J.R., A.S., and N.Y. participated in writing the manuscript.

Ethics declaration

K.L.N. is an employee of BioLegend Inc. The other authors declare no competing interests.

Introduction

The advance of technologies for quantitative, high-throughput measurement of the molecular composition of single cells is continuously expanding our understanding of cell ontology, state, and function [1–3]. A growing body of single-cell multi-omic techniques now offers the ability to further refine our definitions of cellular identity by providing multiple views of molecular state [4, 5]. By extending single-cell RNA-sequencing (scRNA-seq) to simultaneously measure the abundance of proteins on the cell surface, CITE-seq [6,7] presents the opportunity to connect the information that can be gleaned from the transcriptome [8, 9] to the functional information contained in proteins [10, 11]. Such experimental tools necessitate computational tools to synthesize these high-dimensional views.

Recent studies have analyzed CITE-seq data using standard workflows for one modality (often RNA) to cluster cells while contextualizing these results using information from the other modality post-hoc [12–14]. This sequential approach biases the analysis to one modality and becomes increasingly inefficient as CITE-seq measurements expand to hundreds of proteins. A joint analysis that combines these two cellular views in an unbiased manner can harness the strengths of each modality and streamline data analysis. However, combining RNA and protein information to define a single representation of cell state poses several challenges. First, the RNA and protein data have unique sources of technical bias and noise. While the technical aspects of the RNA data have been addressed by a flourishing body of computational methods [15–18], the protein data present distinct technical bias such as background due to ambient or non-specifically bound antibodies. Second, as large-scale community efforts such as the Human Cell Atlas (HCA) [8] begin to include CITE-seq datasets, the need arises for scalable computational methods that can integrate datasets with different measured proteins.

Here, we present totalVI (Total Variational Inference), a deep generative model that enables multifaceted analysis of CITE-seq data and addresses these challenges. totalVI learns a joint probabilistic representation of the paired measurements that accounts for the distinct noise and technical biases of each modality, as well as batch effects. For RNA, totalVI uses a modeling strategy similar to our previous work (scVI; [15]). For proteins, totalVI introduces a new model that separates the protein signal into background and foreground components, which enables background correction. The probabilistic representations learned by totalVI are built on a joint low-dimensional representation of the RNA and protein data that is derived using neural networks. totalVI can be used for disparate analysis tasks including joint dimensionality reduction, dataset integration (with and without missing proteins), protein background correction, estimation of correlations between genes and/or proteins, and differential expression testing. To highlight this functionality, we performed CITE-seq on murine spleen and lymph nodes, measuring up to 208 proteins. We used these data, along with public datasets, to evaluate totalVI's performance across these tasks.

Results

The totalVI model

totalVI uses a probabilistic latent variable model [19] to represent the uncertainty in the observed RNA and protein counts from a CITE-seq experiment as a composite of biological and technical sources of variation. The input to totalVI consists of the matrices of RNA and protein unique molecular identifier (UMI) counts (Fig. 1a). Categorical covariates such as experimental batch or donor are optional inputs used for integrating datasets and referred to henceforth as “batch”. Input datasets can have different antibody panels, and a subset can be scRNA-seq datasets (i.e., without proteins).

The output of totalVI consists of two components that can be used for downstream analysis (Fig. 1b). The first component encodes each cell as a distribution in a low-dimensional latent space (20 dimensions throughout; Supplementary Note 1) that represents the information contained in both the RNA and protein data (Supplementary Note 2), while controlling for their respective noise properties and batch effects. The second component provides a way to estimate the parameters of the distributions that underlie the observed RNA and protein measurements (i.e., likelihoods) given a cell’s latent representation. These distributions explicitly account for nuisance factors in the observed data such as sequencing depth, protein background, and batch effects (Supplementary Note 3). Both components use neural networks to specify distributions.

totalVI optimizes the parameters of both of its components simultaneously using the variational autoencoder (VAE) framework [20]. Accordingly, totalVI uses highly efficient techniques for stochastic optimization that make it appropriate for the scale of CITE-seq data. Following optimization, totalVI’s components are used for downstream analysis. The latent cell representations can be used as input to methods that stratify cells like clustering, visualization, or pseudotime inference algorithms, thus allowing these methods to leverage both protein and RNA information. Other downstream tasks specific to genes and proteins, like differential expression, are linked to the likelihood parameters from the second component of totalVI. Finally, by constricting the latent space to the standard simplex, the dimensions of the latent space can be related to the expression of genes and proteins with archetypal analysis [21], adding an alternative way to investigate global and local patterns of variation in the data. A detailed specification of the model along with further description of the quantities used in downstream tasks is in Methods.

CITE-seq profiling of murine spleen and lymph nodes

We conducted a series of CITE-seq experiments that were designed to test the performance of totalVI on a variety of tasks. As a case study, we profiled murine spleen and lymph nodes, which contain heterogeneous immune cell populations that are well-characterized by surface protein markers. Cells were collected from two wild-type mice that were processed on separate days to serve as biological replicates (Methods). In each experimental run, cells from one mouse were stained with two different panels of barcoded antibodies containing either 111 or 208 antibodies, of which the 111 antibodies were a subset (Supplementary Data). Spleen and lymph node cells stained separately with the same antibody panel were

combined using hashtag antibodies [22]. We refer to the four resulting spleen/lymph node datasets by their panel and experimental day (experimental design in Supplementary Table 1), After pre-processing and filtering, these datasets contained a total of 32,648 cells (Methods).

totalVI fits CITE-seq data well and is scalable

The usefulness of probabilistic models like totalVI depends on how well they fit the observed data. Furthermore, they should generalize to unobserved data (i.e., not overfit) and scale to a realistic range of input sizes. To verify that totalVI satisfies these prerequisites, we benchmarked it against factor analysis (FA), which can be viewed as a linear-Gaussian baseline, scHPF [16], which performs a Poisson matrix factorization via a hierarchical Bayesian model, and scVI [15], which was restricted to the RNA portion of the data. We expected the performance of totalVI and scVI to be comparable on the RNA data, as they share similar architectures. Our evaluation relied on fitting the models to several CITE-seq datasets spanning different species and tissues, including peripheral blood mononuclear cells (PBMC10k) [23] and mucosa-associated lymphoid tissue (MALT) [24] from humans, and our murine spleen and lymph node data (SLN111-D1).

We first estimated how well each model fit data that was available to it during training using posterior predictive checks (PPC) [16, 25]. To conduct PPCs, we generated replicated datasets (i.e., posterior predictive samples) by sampling from the fitted model (Methods). We then assessed how well these replicated datasets maintained the properties of the observed data with two metrics. First, we measured the similarity between the coefficient of variation (CV) per gene and protein of the replicated data to the observed CVs, thus evaluating how well the mean-variance relationship of the data is preserved. Second, we compared the replicated and raw data at the gene and protein level using the Mann-Whitney U statistic, which measures the extent to which the replicated and raw data come from the same distribution. totalVI had superior performance on both metrics (Extended Data Fig. 1a, b).

We then evaluated how well each model generalizes to cells that were not available during training by generating replicated datasets conditioned on the held-out cells and computing two opposing metrics of predictive performance. First, we assessed how well the average replicated data set matched the observed held-out data by mean absolute error. Second, we quantified how well the interval of values from replicated data sets covered the observed held-out data values (calibration error [26]). These two metrics were computed separately for genes and proteins. On the held-out protein data, totalVI outperformed FA in both the mean absolute error and calibration error metrics. Comparing totalVI to scHPF revealed a tradeoff between calibration and held-out error for both the RNA and protein data. On the held-out RNA data, totalVI and scVI were largely comparable and outperformed FA (Extended Data Fig. 2a, b). totalVI and scVI also had a comparable held-out predictive log-likelihood for the RNA data (Extended Data Fig. 2c). Finally, totalVI's performance was also stable across multiple initializations (Extended Data Fig. 2d, e).

To assess the scalability of totalVI, we concatenated all of our spleen and lymph node data (SLN-all) and recorded the training time for different sizes of subsets of this data. totalVI and scVI had similar dependence between run time and input size (Extended Data Fig. 2e).

Furthermore, we observed that totalVI can readily handle large data sets, for instance, processing the complete set of approximately 33,000 cells with over 4,100 features (genes and proteins) in under one hour.

totalVI identifies and corrects for protein background

To analyze protein data in an accurate and quantitative manner, it is necessary to distinguish between true biological signal and technical bias in the protein measurement. Background is a type of technical bias that is characteristic of antibody-based measurements [6, 7, 27]. In CITE-seq data, protein measurements include non-negligible background that arises experimentally from a combination of ambient antibodies, which can be detected in empty droplets, and non-specific antibody binding, which can be detected above ambient levels in cells with no expected expression of a protein, such as CD19 in T cells (Methods, Extended Data Fig. 3a–c, g). Recent methods have described background from ambient RNA [28–30], but the presence of background is more pronounced in protein measurements (Extended Data Fig. 3d–f, Supplementary Note 3).

Previous studies of CITE-seq data derived a single decision rule for every protein, specifying the minimum number of counts required to be considered foreground by using either spiked-in negative control cells [6] or a Gaussian mixture model (GMM) to distinguish a background and foreground component for each protein [31]. Using the same boundary for all cells, however, relies on the assumption that all cells are subject to a similar background distribution of the protein in question and, in the case of a two-component GMM, that the foreground component is comparable across cell types.

totalVI instead models protein background as cell- and protein-specific. To do this, totalVI models each protein measurement as a mixture of foreground and background components that depends on the cell's representation in the latent space, and therefore the full transcriptomic and proteomic profile of that cell. The mixture is weighted by the probability that the counts of a protein in a given cell came from the background component (Fig. 1b, Methods).

To evaluate totalVI's ability to quantitatively identify protein background, we tested how well major cell types could be predicted by the foreground probability (one minus the background probability) of common marker proteins in the SLN111-D1 dataset (Methods). As a baseline for comparison, we used the assignment probabilities from a two-component GMM. For nine out of eleven known marker proteins, both totalVI and the GMM performed well at classifying cell types by marker foreground probability (ROC AUC > 0.97; Supplementary Table 2). For these proteins, such as the B cell marker CD19, the distributions of foreground and background expression were easily separated (Extended Data Fig. 3a and Supplementary Fig. 1a–d). However, for the B cell marker CD20 and the T cell marker CD28, distributions of foreground and background expression were highly overlapping (Extended Data Fig. 3b, c), and totalVI noticeably outperformed the GMM (Extended Data Fig. 3h). totalVI also performed better at distinguishing foreground and background for this set of proteins in the SLN208-D1 dataset, even after normalizing the raw data using isotype control antibodies [32] prior to fitting the GMM (Methods, Supplementary Table 3). Across all proteins, the totalVI foreground probability tended to

fall near zero or one, indicating the model's certainty about most measurements (Supplementary Fig. 1e).

Using CD20 and CD28 as examples, we see how totalVI's identification of protein foreground and background is more accurate than a single decision boundary. In the case of CD20 (encoded by Ms4a1 RNA), a GMM-based cutoff resulted in numerous false negatives (blue cells in Fig. 2a–c, Methods). These cells, identified by totalVI as having high foreground probability despite low CD20 expression, clustered with B cells and expressed Ms4a1 RNA, confirming their identity as B cells. In contrast, cells with similarly low CD20 expression but with low totalVI foreground probability (green cells) clustered with T cells and did not express Ms4a1 (Fig. 2a–c). In the case of CD28, a GMM-based cutoff resulted in numerous false positives (red cells in Fig. 2d–f), while totalVI correctly identified that these cells with high CD28 had low foreground probability, and were in fact B cells rather than T cells. totalVI is not limited to distinguishing globally bimodal distributions (e.g., CD4 in peripheral blood mononuclear cells globally follows a trimodal distribution (Methods, Extended Data Fig. 4a, b)).

For downstream analysis, totalVI uses foreground probabilities in a quantitative manner to remove protein background. Specifically, totalVI can denoise the protein data by setting the background component to zero, while also accounting for the measurement uncertainty of the foreground component (Methods, Fig. 2g–j, Extended Data Fig. 4f, g). We use the expectation of denoised values for visualization (Extended Data Fig. 4c–e).

For statistical analyses like differential expression testing, totalVI uses distributions over the denoised values as opposed to testing directly on a denoised data matrix, which could introduce bias [33]. For analyses focused on the relationships between features, we developed a novel sampling method that controls for nuisance variation while avoiding denoising-induced artifacts (Methods). We applied this method to construct denoised feature-feature correlation matrices and found that totalVI preserved the independence of negative control genes (Extended Data Fig. 5a, b, d, e), lending confidence that downstream analysis with totalVI is not subject to spurious feature relationships arising from data denoising. Observing the correlations between proteins and their encoding RNA, we found that totalVI correlations were generally higher in magnitude than raw correlations (Extended Data Fig. 5c, f).

totalVI integrates CITE-seq datasets

We next evaluated totalVI's ability to integrate data from CITE-seq experiments that measured different sets of proteins. Integration is built into totalVI via an assumption of independence between the latent space and the batch. Consequently, totalVI produces both an integrated latent space, as well as corrected expression values. In the case of unmatched protein panels, totalVI can impute missing proteins for a particular dataset by using the information learned from those proteins in the datasets in which they were observed (Methods). We applied totalVI to the SLN111-D1 and SLN208-D2 datasets, which had a clear batch effect that was revealed by principal component analysis (Fig. 3a). We benchmarked totalVI against three state-of-the-art integration methods: Seurat v3 [34], Scanorama [35], and Harmony [36]. We assessed totalVI in the case of matched panels

(using only the 111 overlapping proteins between the two panels; denoted as totalVI-intersect) and unmatched panels (using the union of the two protein panels, which results in missing data for some proteins; denoted as totalVI-union). Despite being designed for scRNA-seq, the other methods could be extended to handle CITE-seq data, though only in the case of matched panels (Methods).

We used four metrics to quantify how well each method mixed datasets along with how well they maintained the original structure of each dataset (Methods). The first two metrics (the latent mixing metric and the measurement mixing metric) quantify how well cells mix across datasets in the low-dimensional latent space and the observed expression space (per feature), respectively. The second two metrics (the feature retention metric and clustering metric) summarize how well each method preserves each dataset's original structure, either at the feature-level through autocorrelation (feature retention metric), or at the cell-level through clusters (clustering metric). Finally, we benchmarked totalVI's accuracy of predicting protein expression in cases where measurements are available in only one of the datasets.

We found that after integration, cells of similar types were co-located in the latent space, as evidenced by the shared expression of key marker proteins like CD4, CD8a, and CD19 (Fig. 3b, c; Supplementary Fig. 2). Moreover, totalVI outperformed the other methods in the feature retention and clustering metrics, while comparing favorably in the remaining metrics (Fig. 3d, e). totalVI-union and totalVI-intersect performed similarly, indicating that the presence of missing data did not diminish totalVI's integration capabilities. We repeated this analysis on two public datasets of PBMCs (PBMC10k [23], PBMC5k [37]), which also had very different sequencing depths, and observed similarly favorable performance for totalVI (Supplementary Fig. 3a–f).

Since totalVI-union can integrate CITE-seq datasets with different protein panels, we reasoned it could also integrate a CITE-seq dataset with a standard scRNA-seq dataset that has not measured proteins and impute the missing protein measurements. We assessed this by integrating SLN111-D1 and SLN111-D2, where we held out the proteins of SLN111-D2. We first observed that totalVI can learn a biologically meaningful integrated latent representation despite the large amount of missing data (Fig. 3f). Indeed, the location of observed protein expression in the latent space revealed the same broad immune cell types. Next, we imputed the protein expression for the cells in SLN111-D2 (Methods). For key cell type marker proteins, totalVI-imputed proteins shared similar patterns of expression as the held-out observed proteins (Fig. 3g).

To further quantify imputation accuracy, we ran totalVI 30 times with resampled training sets and, for each run, computed the root mean squared log error between imputed and observed protein values. We compared totalVI to Seurat v3, which imputes protein values based on smoothing of protein values from mutual nearest RNA neighbors. The accuracy of 80 proteins was significantly different between totalVI and Seurat v3 (Student's T-test, Benjamini–Hochberg (BH)-adjusted p-value <0.05). The mean error of totalVI was better than the Seurat v3 error for approximately 68% of the 80 proteins (Fig. 3h). We also performed this task on PBMCs (Supplementary Fig. 3h, i), in which we also compared to

another protein imputation method, cTP-net [38]. We found that totalVI and Seurat v3 performed more similarly, while outperforming cTP-net. For further discussion on the merits and limitations of imputing missing proteins, see Supplementary Note 4.

totalVI identifies differentially expressed genes and proteins

totalVI can leverage its estimates of uncertainty from a single model fit to detect differentially expressed features between two sets of cells while controlling for noise and other modeled technical biases like sequencing depth (RNA), background (protein), and batch effects (both). To do so, totalVI estimates a distribution over the log fold change (LFC) of expression between the two sets of cells, which is then used to quantify how well the data support a hypothesis of differential expression (using Bayes factors [15, 39, 40]; Methods).

To evaluate totalVI as a framework for differential expression (DE) analysis in the common scenario of multiple experiments, we integrated all four spleen and lymph node datasets (SLN-all; totalVI-intersect). totalVI provided a descriptive representation of this data, as inspection of established cell type markers associated clusters of cells in the latent space with immune cell types or states (Fig. 4a, Extended Data Fig. 6, Methods).

Beyond markers used for annotation, we found that a totalVI one-vs-all DE test (in which one cell type is compared to all others) identified many additional features as differentially expressed (Methods, Fig. 4b, c; Supplementary Data). For example, totalVI identified the gene Klrc2 as differentially expressed in both natural killer (NK) cells and gamma/delta T cells, which has previously been shown to be upregulated in these populations relative to alpha/beta T cells [41]. For proteins, totalVI identified CD335 (NKp46) as among the top markers for NK cells, which is a canonical marker used for sorting [42], and CD43, which is associated with the development of NK cells [43].

Overall, the Bayes factors inferred by totalVI for the RNA data were highly correlated with those produced by scVI (Extended Data Fig. 7a), which has been independently evaluated [40]; therefore, we focused on evaluating the protein DE test. Throughout, we compared totalVI to two baseline methods: a Welch's t-test and a Wilcoxon rank-sum test. We also compared to a version of totalVI in which the protein background was not corrected (totalVI-wBG).

We first evaluated the extent of false positives using isotype control antibodies. As isotype controls lack target specificity, differences in their abundance between cell types likely stem from background or other technical sources of variation. Applying each method to the SLN208-D1 dataset, which contained nine isotype controls, we found that totalVI called the fewest (and often zero) isotype controls as differentially expressed in one-vs-all tests (Extended Data Fig. 7b). We next tested the reproducibility of the methods across biological replicates, finding that totalVI outperformed the baseline methods (Extended Data Fig. 7c–e). The totalVI DE test was also reproducible across experimental designs: one in which the two CITE-seq datasets had the same protein panel, and another in which proteins were measured in only one of the datasets (Extended Data Fig. 7f).

To gain further insight into the extent of false positive and false negative DE calls, we compared ICOS-high regulatory T cells (ICOS-high Tregs) and conventional CD4 T cells from SLN-all. This test is challenging because these two cell types share many of the same upregulated and downregulated features when compared with other immune cell types. Our analysis was based on a list of putative positive and negative surface proteins curated from previous studies that used flow cytometry (Methods).

We found that totalVI and the baseline methods identified these putative positives as significantly upregulated; however, the two baseline methods also incorrectly called all putative negatives as upregulated (Fig. 4d). Globally, the two baseline methods both called 78 out of 110 proteins as differentially expressed, many of which are likely the result of differences in background. While filtering proteins by the observed LFC in the baseline methods may reduce these false positives, the improvement would be limited (e.g., CD5 and IgD had similar LFCs and therefore could not be distinguished; Fig. 4d). The totalVI test, in contrast, correctly classified all putative negatives and positives (Fig. 4e), calling 28 proteins differentially expressed in total. To further support the utility of correcting for protein background, we performed this test using totalVI-wBG, which improved upon the baseline methods, but also falsely called some putative negatives as positives (Supplementary Fig. 4a).

Finally, totalVI's LFC estimates (defined as the median of the LFC distribution) better captured the underlying biological signal. For example, in a test of CD4 T cells vs all from SLN-all, the canonical marker CD4 had a higher LFC than in the raw data (Fig. 4f). Additional markers like CD28 (T cell marker) and CD20 (B cell marker), which we previously highlighted as having highly overlapping foreground and background components, had respectively higher and lower LFCs compared to LFCs derived from the raw data.

totalVI provides an interpretable latent space

Deep-learning-based methods for dimensionality reduction tend to rely on "black-box" models, making it difficult to interpret the coordinates of their inferred low-dimensional latent spaces. Despite the non-linear relationship between the totalVI latent space and the expression space, totalVI provides a way to relate each latent dimension to the expression of individual features via archetypal analysis [21, 44, 45] (Methods). Archetypes, which correspond to dimensions of the latent space, represent a summary of expression programs, the combination of which characterizes a cell. To demonstrate archetypal analysis, we ranked the features most associated with each archetype in the SLN-all dataset (Extended Data Fig. 8a, b), finding that some archetypes corresponded to specific cell types, and others captured more global variation (Extended Data Fig. 9a). For example, archetype 16 was associated with high protein expression of CD93 and CD24, which mark the transitional B cell subset (Extended Data Fig. 9b). In contrast, archetype 7 was associated with interferon-response genes such as Ifit3 and Isg20 and reflected within-cell-type variability in several subsets, including CD4 and CD8 T cells, B cells, Ly6-high monocytes, and neutrophils (Extended Data Fig. 9c and Supplementary Fig. 5). We also used archetypal analysis to evaluate the influence of proteins on the latent space, and found that all but one archetype

had proteins overrepresented in its top features (Extended Data Fig. 8c). This suggests that the inclusion of proteins significantly influences representations in the totalVI latent space.

Characterization of B cell heterogeneity in the spleen and lymph nodes with RNA and proteins

We next demonstrate how a joint representation of RNA and protein can be used to characterize cell identities within a specific immune compartment and in the context of multiple samples. Here, we used the totalVI-intersect model fit on the SLN-all dataset and focused on the B cell population (Methods, Fig. 4a).

We started with characterizing cell identities using prior biological knowledge by visualizing the expression of six surface proteins commonly used for isolating B cell subsets (Fig. 5b, Supplementary Table 4). These subsets included transitional (marked by CD93 and CD24), mature (marked by IgD and CD23), B1 (marked by CD43) and marginal zone (MZ, marked by CD21) B cells. These markers stratified the B cells into groups that were largely consistent with unsupervised clustering (Methods). RNA expression of these markers followed similar patterns to the proteins they encode (Fig. 5c).

The difference in subset composition between the spleen and lymph nodes (Fig. 5d) was consistent with previous studies (Fig. 5e, [46, 47]). In particular, clusters spanned the developmental range from recent bone-marrow emigrants in the splenic transitional B cell subset to mature cells present in both tissues. As expected, the B1 and MZ B cell subsets were found primarily in the spleen.

In a more unbiased approach, we quantified the differences between the B cell clusters with the totalVI one-vs-all DE test (Fig. 5f, g, Methods). As expected, the six known surface markers were among the top differentially expressed protein markers (Fig. 5f). Most RNA molecules encoding the marker proteins were also differentially expressed along with informative genes whose products are not present on the cell surface, such as the transcription factor Bhlhe41 that marks B1 B cells (Fig. 5g, [48]).

Globally, protein data combined with a transcriptome-wide view enabled a more refined characterization of variation within the four major sub-populations identified above by surface markers. For example, a sub-population of mature B cells labeled here as Ifit3-high B cells expressed all of the protein and RNA markers of mature B cells and could not be clearly distinguished from the remaining mature B cells based on protein data alone (maximum LFC across all proteins was less than 0.19). Nevertheless, with transcriptome-wide DE analysis, this cluster could be distinguished as a sub-type of mature cells by the elevated expression of interferon response genes (Fig. 5g). This observation was supported by a gene signature analysis with Vision [49], which identified two interferon response signatures enriched in the Ifit3-high B cell cluster (Methods, Supplementary Fig. 5a, b). The expression of interferon response genes was not expected since no inflammation was induced, however we found the Ifit3-high B cell cluster as well as Ifit3-high T cell clusters to be represented in both biological replicates, and therefore took it to capture part of the biology in the SLN-all dataset (Supplementary Fig. 5c, d).

Next, we explored the variability within transitional B cells and its relationship with B cell development. Interestingly, latent dimension 16 (Z_{16}) captured a gradual transition within this cluster: from a small population of Rag1 expressing cells (indicating early development [46]) to cells that were closer to the mature cluster (Fig. 5i, Extended Data Fig. 10a, b). To explore how development from transitional to mature B cells may be associated with coordinated changes in gene and protein expression, we calculated the totalVI Spearman correlations separately within transitional and mature B cells for a set of features that distinguished the two subsets (Methods). Hierarchical clustering of the correlation matrix within the transitional B cells clearly stratified these features into two anti-correlated modules: one associated with transitional B cells and the other with mature B cells (Fig. 5h). These modules, however, were not present in mature B cells, indicating that the apparent coordination may be a characteristic of the transitional state (Extended Data Fig. 10c). Within transitional B cells, we found that the features in the two modules significantly correlated with the axis of maturation captured by Z_{16} (Extended Data Fig. 10d). Along this axis, features in the transitional module decreased while those in the mature module increased (Fig. 5j, Methods). These results point to a program of transitional B cell maturation that consists of coordinated activation and repression of multiple genes and proteins, leading to a gradual transition in cell state that is captured by a specific dimension of the totalVI latent space.

Discussion

totalVI is a scalable, probabilistic framework for end-to-end analysis of paired transcriptome and protein measurements in single cells. Like other multi-omics analysis methods [31, 50, 51], totalVI assumes that RNA and protein measurements are generated from the same latent space of cells that captures their state. A distinction of totalVI is that it explicitly models modality-specific technical factors like protein background, which we demonstrated can enable a denoised view of the data and more accurate differential expression results. totalVI is also unique in its ability to handle missing protein data, which enables integration with growing public data resources like the Human Cell Atlas [8].

Beyond the characterization of cell types, totalVI can also uncover relationships between RNA and protein molecules within a cell. For example, totalVI could be used to investigate the relationship between the level of an RNA transcript and the level of its encoded protein in different biological settings, which remains an open question [52]. We found that the totalVI correlations were higher in magnitude than raw correlations across the majority of RNA-protein pairs, suggesting that the low correlations observed previously [6, 7] could have been due to technical noise. Future work quantifying correlations and regulatory relationships between RNA and protein features could inform our understanding of signal transduction pathways or transcription and translation dynamics [53].

While the totalVI model was designed to reflect our understanding of the CITE-seq experimental data-generating process (Supplementary Note 3), totalVI can also be used to inform experimental design. For instance, totalVI could help identify antibody titrations or experimental methods that improve signal-to-noise. totalVI could also identify sequencing

depths for RNA and protein libraries that balance the information gained per measurement in various analysis tasks with the cost of additional sequencing [54, 55].

Through a single pipeline that jointly analyzes paired RNA and protein measurements, totalVI simplifies data analysis and interpretation that would otherwise be conducted in separate pipelines whose disparate results must be reconciled post hoc. totalVI is available through the scvi-tools software package, which connects it with the popular Scanpy [56] and Seurat [34] pipelines, and enables analysis on free cloud computing environments like Google Colab. The flexibility and scalability of totalVI make it easily applicable to future datasets with larger protein panels, and enable extensions that incorporate additional paired measurements. For example, we expect totalVI to naturally handle intracellular proteins measured with barcoded antibodies. Further additions of modalities like chromatin accessibility [57] or clonotype features [58] can also be implemented within the totalVI codebase with consideration of the modality-specific likelihood. By combining multiple views of cellular processes, totalVI can reveal a more complete picture that redefines cell states and elucidates mechanistic relationships between molecular components of the cell.

Methods

The totalVI model

totalVI estimates a conditional distribution for cell n, $p_v(x_n, y_n | s_n)$, in which x_n is the G-dimensional vector of observed RNA counts (G genes), y_n is the T-dimensional vector of observed protein counts (T proteins) and s_n is the B-dimensional one-hot vector describing the batch index (experiment identifier). In total, there are N cells. We use v to refer to the set of all generative parameters, which are described throughout this section. This distribution is estimated using the framework of variational autoencoders (VAE; [20]).

We begin by describing the generative process, for which a graphical summary is in Supplementary Fig. 6 and an algorithmic summary is in Algorithm 1. We then describe the inference procedure, as well as how downstream analysis tasks are directly linked to posterior queries of the model.

Priors—The latent cell representation $z_n \sim \text{LogisticNormal}(0, I)$, where the logistic normal distribution is a distribution over the probability simplex. This specification, which has also been applied in the context of linear VAEs for scRNA-seq [59], enables cells to be interpreted with archetypal analysis. Typically in VAEs, z_n follows an isotropic normal distribution, which is chosen for computational convenience [20]. In this setting, a logistic normal distribution arises as transforming a sample from a normal distribution with a softmax function. For all experiments, we set z_n to 20 dimensions. We discuss the choice of number of latent dimensions in Supplementary Note 1.

The latent RNA size factor $\ell_n | s_n \sim \text{LogNormal}(\ell_\mu^\top s_n, \ell_\sigma^\top 2 s_n)$, where $\ell_\mu \in \mathbb{R}^B$ and $\ell_\sigma^2 \in \mathbb{R}_+^B$ are set to the empirical mean and variance of the log RNA library size (defined as total RNA counts of a cell) per batch. We use a protein-specific prior for the protein background

intensity, where $\beta_{nt} \mid s_n \sim \text{LogNormal}(c_t^\top s_n, d_t^\top s_n)$. The parameters for the background intensity, $c_t \in \mathbb{R}^B$ and $d_t \in \mathbb{R}_+^B$, are protein specific and are treated as model parameters learned during inference. This prior is motivated by the observation that some component of the background is due to ambient antibodies. By being batch specific, these priors on ℓ_n and β_n account for differences in sequencing depth between datasets. A prior can also be thought of as regularizing the posterior distribution, thus reducing the influence of outliers [60]. The selection of prior distribution was guided by the computational tractability and by properties that are of interest (e.g., non-negativity).

RNA likelihood—Given z_n, ℓ_n, and s_n, an observed expression level x_{ng} follows a negative binomial distribution, which we present here as a Gamma-Poisson mixture:

$$\rho_n = f_\rho(z_n, s_n) \tag{1}$$

$$w_{ng} \mid z_n, \ell_n, s_n \sim \text{Gamma}(\theta_g, \ell_n \rho_{ng}) \tag{2}$$

$$x_{ng} \mid w_{ng} \sim \text{Poisson}(w_{ng}) \tag{3}$$

The gamma distribution is parameterized by its shape and mean. The mean is equal to $\ell_n \rho_{ng}$, where ℓ_n, a scaling factor, is multiplied by ρ_{ng}, interpreted as a normalized gene frequency (because ρ_n is nonnegative and sums to one). ρ_n is the output of a neural network f_ρ, which takes z_n and s_n as input (Algorithm 1).

Integrating out w_{ng} results in the following conditional distribution:

$$x_{ng} \mid z_n, \ell_n, s_n \sim \text{NegativeBinomial}(\ell_n \rho_{ng}, \theta_g) \tag{4}$$

The parameter θ_g, which is the shape of the gamma distribution, is also the inverse dispersion of the negative binomial (Supplementary Note 5). We perform inference on the model with w_{ng} integrated out. We also treat θ_g as a model parameter learned during inference. Overall, this likelihood is equivalent to that presented in scVI [15], without zero-inflation. The negative binomial distribution has been shown to adequately handle the limited sensitivity and over-dispersion that are characteristic of this data [61].

Protein likelihood—To capture observed protein counts arising from the background or foreground, we model y_{nt} with a negative binomial mixture, given z_n, β_n and s_n. This conditional distribution is described by the following process:

$$\pi_n = h_\pi(z_n, s_n) \tag{5}$$

$$\alpha_n = g_\alpha(z_n, s_n) \tag{6}$$

$$v_{nt} \mid z_n, s_n \sim \text{Bernoulli}(\pi_{nt}) \tag{7}$$

$$r_{nt} \mid v_{nt}, \beta_{nt}, z_n, s_n \sim \text{Gamma}(\phi_t, v_{nt}\beta_{nt} + (1 - v_{nt})\beta_{nt}\alpha_{nt}) \tag{8}$$

$$y_{nt} \mid r_{nt} \sim \text{Poisson}(r_{nt}) \tag{9}$$

Here v_{nt} controls which mixture component generates the counts. Its parameter, π_{nt}, is the output of the neural network $h_\pi(z_n, s_n)$. Notably, α_{nt}, which is the output of the neural network $g_\alpha(z_n, s_n)$, is greater than one. This ensures that one of the mixture components is always larger than the other, allowing us to interpret one component as background and one component as foreground. Furthermore, π_{nt} is interpreted as the probability that any cell-protein pair has observed counts due to background alone. For one mixture component, $y_{nt} \mid z_n, \beta_{nt}, s_n, v_{nt}$ follows a negative binomial distribution, as can be seen by integrating out r_{nt}. Finally, integrating out v_{nt} too shows that y_{nt} given z_n and s_n follows a negative binomial mixture distribution, where ϕ_t is a protein-specific inverse dispersion parameter.

Algorithm 1: The totalVI generative model. The gamma distriution is parameterized by its shape and mean. Let v be the set of model parameters described here. A dataset has G genes and T measured proteins.

Define: Neural networks

$f_\rho(z_n, s_n): \Delta^{K-1} \times \{0,1\}^B \rightarrow \Delta^{G-1}$, (Softmax output activation)

$g_\alpha(z_n, s_n): \Delta^{K-1} \times \{0,1\}^B \rightarrow [1, \infty)^T$, (ReLU + 1output activation)

$h_\pi(z_n, s_n): \Delta^{K-1} \times \{0,1\}^B \rightarrow (0,1)^T$ (Sigmoid output activation)

Require: Inverse dispersion parameters $\theta \in \mathbb{R}_+^G, \phi \in \mathbb{R}_+^T$. Neural network parameters.

for *eachcelln* **do**

$z_n \sim \text{LogisticNormal}(0, I)$ K − dim.cellular state variable

$\rho_n = f_\rho(z_n, s_n)$ G − dim. RNA frequency

$\alpha_n = g_\alpha(z_n, s_n)$ T − dim. foreground increment protein scaling

$\pi_n = h_\pi(z_n, s_n)$ T − dim. mixture parameter

$\ell_n \sim \text{Lognormal}\left(\ell_\mu^\top s_n, \ell_\sigma^\top 2 s_n\right)$ Cell scaling factor for RNA

for *each gene g* **do**

$w_{ng} \sim \text{Gamma}\left(\theta_g, \ell_n \rho_{ng}\right)$

$x_{ng} \sim \text{Poisson}\left(w_{ng}\right)$

for *each protein t* **do**

$\beta_{nt} \sim \text{Lognormal}\left(c_t^\top s_n, d_t^\top s_n\right)$ Scalar background mean

$v_{nt} \sim \text{Bernoulli}(\pi_{nt})$ Scalar mixture assignment

if $v_{nt} = 1$ **then**

$r_{nt} \sim \text{Gamma}(\phi_t, \beta_{nt})$

$y_{nt} \sim \text{Poisson}(r_{nt})$

else

$r_{nt} \sim \text{Gamma}(\phi_t, \beta_{nt}\alpha_{nt})$

$y_{nt} \sim \text{Poisson}(r_{nt})$

Inference for totalVI

Inference in the case of fully observed proteins—The model evidence, $p_v(x_{1:N}, y_{1:N} | s_1 N)$, cannot be computed as the integrals are analytically intractable, so Bayes rule cannot be directly applied to find the posterior distribution. Therefore, we use variational inference [62] to approximate the posterior distribution with a distribution having the following factorization:

$$q_\eta(\beta_n, z_n, \ell_n | x_n, y_n, s_n) := q_\eta(\beta_n | z_n, s_n) q_\eta(z_n | x_n, y_n, s_n) q_\eta(\ell_n | x_n, y_n, s_n). \quad (10)$$

Here η is the set of parameters of an inference network, commonly called the *encoder* – a neural network that takes a cell's combined expression as input and outputs the parameters of the approximate posterior (e.g., mean and variance). Factors of the posterior approximation share the same family as their respective priors (e.g., $q(\beta_n | z_n, s_n)$ is lognormal). The approximate posterior $q_\eta(z_n | x_n, y_n, s_n)$, whose expectation we use as the latent cell representation, is integral to many cell-level and feature-level analyses.

For the likelihoods, as described previously, we integrate out the latent variables v_{nt}, r_{nt} and w_{ng} (Algorithm 1), yielding $p_v(y_{nt} | z_n, \beta_{nt}, s_n)$, which is a mixture of negative binomials and $p_v(x_{ng} | z_n, s_n, \ell_n)$, which is a negative binomial distribution.

The evidence lower bound (ELBO) [62] of $\log p_v(x_{1:N}, y_{1:N} | s_{1:N})$ is optimized with respect to the variational parameters η and model parameters ν using stochastic gradients [20]. In other words, the model parameters and approximate posterior parameters are learned simultaneously. In the VAE framework, the generative neural network is referred to as the *decoder*. Each iteration of training consists of randomly choosing a mini-batch of data (256 cells), estimating the ELBO based on this mini-batch, and updating the parameters via automatic differentiation operators. The terms corresponding to Kullback-Leibler divergences of the ELBO (Supplementary Note 6) follow a deterministic warm-up scheme [63], which helps to avoid shallow local maxima. We use the Adam optimizer [64] with weight decay to update the model parameters. Learning rate reductions and early stopping are performed based on the ELBO of a validation set. As a result of mini-batching, totalVI's memory usage is constant in the number of features in the dataset and number of neural network parameters. For example, in the runtime experiment presented in Extended Data Fig. 2f, totalVI used a constant 753 megabytes of memory on an NVIDIA Titan XP GPU. totalVI's runtime is linear in the number of cells and linear in the number of features; however, as we use early stopping, convergence may vary with the dataset size.

All neural networks are feedforward and use standard activations (e.g., exponential, softmax, sigmoid) to encode the variational and generative distributions. We use the same hyperparameters for all of our experiments. Supplementary Note 6 gives further implementation details.

Inference in the case of missing proteins—Here we adapted the training procedure from [65] to handle missing protein data. As any single batch may correspond to an experiment that used a different protein panel (or no proteins in the case of a scRNA-seq experiment), the missingness of protein features depends on the batch index s_n. Further, suppose all batches share the same set of genes. Across all batches, there are T proteins. For cell n, we denote the observed protein expressions y_n^{obs} and the unobserved protein expressions y_n^{mis}. The log likelihood of the observed data decomposes as

$$\log p_v(x_{1:N}, y_{1:N}^{\text{obs}}, s_{1:N}) = \sum_{n=1}^{N} log p_v(x_n, y_n^{\text{obs}} | s_n) \tag{11}$$

The generative process for the observed data is the same as in Algorithm 1, with appropriate modification to only generate the features present in a particular batch. Thus, ν is the same set of model parameters described previously. Again, we use variational inference to approximate the posterior distribution with the distribution in Equation 10. In fact, all approximate posteriors share the same encoder parameters η. We optimize the ELBO of Equation 11 similarly to the procedure used when there is no missing data (i.e., we optimize the ELBO with respect to the model parameters ν and variational parameters η). To handle

mismatched dimensions in the encoder, we substitute zeros for missing proteins, and for the decoder, we only calculate the ELBO terms corresponding to observed data [66]. Therefore, this procedure naturally extends to the case when there is no observed protein data for a cell n, which would be the case when the cell is obtained from a scRNA-seq experiment. Since the quality of missing protein imputation depends on (i) the goodness of fit of totalVI to the protein for the data in which it was observed and (ii) the statistical distance of the aggregated posterior distributions of z_n for each of the batches [65, 67], we add a domain adaptation regularization term to the ELBO when training [68]. A scaling factor on this regularization term decays from one to zero early in training.

Posterior predictive distributions linked to downstream tasks

For tasks like differential expression, denoising, and finding correlations, totalVI estimates functionals of posterior predictive distributions [19]. Define $C_n = \{x_n, y_n, s_n\}$ as the set of observed data for cell n. First, consider the connection between the posterior predictive distribution of RNA data to totalVI denoised RNA expression. The posterior predictive RNA expression x^*_{ng} for gene g given C_n is distributed following:

$$p\left(x^*_{ng} \middle| C_n\right) \approx \int p_v\left(x^*_{ng} \middle| z_n, l_n, s_n\right) q_\eta(z_n, l_n | C_n) dz_n dl_n, \tag{12}$$

To produce denoised RNA expression, we compute the posterior predictive mean of x^*_{ng}. To further control for variation due to ℓ_n, we condition on $\ell_n = 1$. By the law of total expectation,

$$\mathbb{E}_p(x^*_{ng} \middle| x_n, \ell_{n=1})[x^*_{ng}] = \mathbb{E}_{q_\eta(z_n|C_n)}\left[\mathbb{E}_{p_v(x^*_{ng}|z_n s_n, \ell_{n=1})}[x^*_{ng}]\right] \tag{13}$$

$$= \mathbb{E}_{q_\eta(z_n|C_n)}[\rho_{ng}], \tag{14}$$

where ρ_{ng} is the expectation of the RNA likelihood with the additional condition that $\ell_n = 1$.

For each cell n, we can compute the denoised RNA expression by averaging samples of ρ_n generated by the following process:

1. Sample z_n from $q_\eta(z_n|C_n)$
2. Set $\rho_n = f_\rho(z_n, s_n)$

There are two important considerations for these posterior predictive distributions. First, we use the approximate posterior as a surrogate for the posterior. Second, these posterior predictive distributions are not tractable to compute in closed form, so we can only sample from them with ancestral sampling. Functionals of the posterior are computed using Monte Carlo integration.

Denoised protein expression—After training the model, we can generate "denoised" protein expression – protein expression effectively absent of background and controlled for sampling noise. Consider the perturbed protein generative process in which we set the background intensity to zero:

$$v_{nt} \mid z_n, s_n \sim \text{Bernoulli}(\pi_{nt}) \tag{15}$$

$$\tilde{r}_{nt} \Big| v_{nt}, \beta_{nt}, z_n, s_n \sim \begin{cases} \text{Gamma}(\phi_t, \beta_{nt}\alpha_{nt}) & \text{if } v_{nt} = 0 \\ \delta_0 & \text{if } v_{nt} = 1 \end{cases} \tag{16}$$

Here δ_0 is a point mass distribution at 0. After marginalizing out v_{nt}, $\tilde{r}_{nt} \mid z_n$, s_n, β_{nt} follows a zero-inflated Gamma distribution with mean $(1 - \pi_{nt})\beta_{nt}\alpha_{nt}$.

For denoising, we return the posterior predictive mean of $\tilde{r}_{nt}$. Indeed, the posterior predictive mean is equal to $(1 - \pi_{nt})\beta_{nt}\alpha_{nt}$ averaged over many posterior samples of $q(\beta_{nt}, z_n \mid C_n)$. In other words, we return the foreground mean, weighted by the probability that the observation was derived from the foreground. This can also be stated as subtracting the expected background from the expected total signal.

Missing protein imputation—To impute protein expression y^*_{nt} for cell n and protein t missing in batch s_n, but that is observed in a batch $s' \neq s_n$, do the following:

1. Sample z_n from $q_\eta(z_n \mid C_n)$
2. Sample β_{nt} from $q_\eta(\beta_{nt} \mid z_n, s = s')$
3. Sample y^*_{nt} from $p_v(y^*_{nt} \mid , z_n, \beta_n, s = s')$

This process returns samples of $p_v(y^*_{nt} \mid C_n, s = s')$. Intuitively, we encode the cell into the latent space, which is designed to mix the batches (i.e., be an integrated low-dimensional representation of the data), and obtain the parameters for the protein likelihood (decode) conditioned on the cell being in batch $s = s'$. Thus, the quality of imputation relies on how well batches mix in the totalVI latent space. Ultimately, we report the expected value of the imputed distribution

$$\mathbb{E}_{p(y^*_{nt} \mid C_n, S = 1)}[y^*_{nt}] = \mathbb{E}_{q_\eta(z_n \mid C_n)}\left[\mathbb{E}_{p(y^*_{n,t} \mid z_n, s = 1, \beta_{nt})}[y^*_{nt}]\right] \tag{17}$$

We may also impute the denoised expression, by exchanging $p_v(y^*_{nt} \mid z_n, \beta_n, s)$ with $p_v(\tilde{r}_{nt} \mid z_n, \beta_n, s)$. This change would additionally remove the protein background contribution to the prediction.

Differential expression—With a single model fit, totalVI can detect differentially expressed features between sets of cells, i.e., the model does not need to be retrained for

every test. Here we use the Bayesian framework of [40] to detect differential expression (DE) of genes and proteins. Let

$$\lambda_{a,b} := \Lambda(z_a, z_b, s_a, s_b) := \log_2\rho_a - \log_2\rho_b \tag{18}$$

be the log fold change (LFC) of RNA expression between cells a and b. Then the probability that gene g is differentially expressed (DE) is

$$p(|\lambda^a_{a,b}| \geq \delta | C_a, C_b) \approx \int 1\{|\lambda^a_{a,b}| \geq \delta\} q(z_a|C_a) q(z_b|C_b) dz_a dz_b, \tag{19}$$

where δ is a threshold for the effect size. Intuitively, we are measuring the fraction of posterior samples that the absolute LFC greater than or equal to δ. For all experiments we set $\delta = 0.2$. We compare the DE probability to the probability that the LFC is in the null region $|\lambda^g_{a,b}| < \delta$ using a Bayes factor:

$$\mathrm{BF}^g_{a,b} = \frac{p(\left|\lambda^g_{a,b}\right| \geq \delta \left| C_a, C_b\right)}{p(\left|\lambda^g_{a,b}\right| < \delta \left| C_a, C_b\right)}. \tag{20}$$

This can also be extended to groups of cells. Let $A = a_1, a_2, \ldots, a_m$ be the indices of one subpopulation of interest, and $B = b_1, b_2, \ldots, b_n$ be the other subpopulation of interest. We then exchange the posterior distributions in Equation 19 with the aggregated posterior:

$$q_\eta(z_a|C_A) q_\eta(z_b|C_B) = \left[\frac{1}{|A|}\sum_{a \in A} q_\eta(z_a|C_a)\right]\left[\frac{1}{|B|}\sum_{b \in B} q_\eta(z_b|C_b)\right]. \tag{21}$$

In this sampling procedure, a cell representation z_a (resp.z_b) is sampled given one randomly chosen cell in subpopulation A (resp. subpopulation B). Then, it is determined if $|\lambda^g_{a,b}| \geq \delta$ via an indicator function. The DE probability is estimated based on many samples.

Furthermore, by integrating over the batch variable s_n, we effectively compare cells as if they were in the same batch [15]. For genes, this is equivalent to computing

$$p(|\lambda^g_{a,b}| \geq \delta | C_a, C_b) \approx \sum_{s'} \int 1\{|[\Lambda(z_a, z_b, s', s')]^g \geq \delta\} p(s') q(z_a|C_a) q(z_b|C_b) dz_a dz_b. \tag{22}$$

Here $p(s')$ is a uniform prior over batches. Every time we sample from the posterior, we decode the samples using the same batch indicator, averaging the DE probability over every possible batch indicator.

For proteins, we use the same framework, but define

$$\gamma^{t}_{a,b} = \log_2(\mathbb{E}[\tilde{r}_{at} | \beta_{at}, v_{at}, z_a] + \epsilon) - \log_2(\mathbb{E}[\tilde{r}_{bt} | \beta_{bt}, v_{bt}, z_b] + \epsilon), \tag{23}$$

where the conditional expectation is equal to

$$\mathbb{E}[\tilde{r}_{at} | \beta_{at}, v_{at}, z_a] = \beta_{at}\alpha_{at}(1 - v_{at}). \tag{24}$$

This is interpreted as the foreground mean if the cell was generated from the foreground, and zero otherwise. The added constant ϵ is a "prior count" that helps define the log fold change when $\mathbb{E}[\tilde{r}_{nt} | \beta_{nt}, v_{nt}, z_n] = 0$. For all analysis, we set $\epsilon = 0.5$. As with genes, we are interested in calculating $p(|\gamma^{t}_{a,b}| \geq \delta | C_a, C_b)$, where in this case we integrate with respect to the distribution

$$\prod_{i \in a,b} p(v_{it} | z_i) q(\beta_{it} | z_i, s_i) q(z_i | C_i). \tag{25}$$

We consider features with a $\log(\text{BF}) > 0.7$ as differentially expressed. This is roughly equivalent to calling features significant if the odds ratio (here equivalent to a Bayes factor) is greater than 2. Finally, we use the posterior samples of $\lambda_{a,b}$ (resp. $\gamma_{a,b}$ for proteins) as the estimate of effect size for each gene (resp. protein). Specifically, we use the median of the samples, which is robust to outliers and is also the Bayes estimator under L_1 loss.

Denoised correlation matrix construction—We seek a feature-feature correlation matrix (e.g., gene-gene correlations, gene-protein cross-correlations) that summarizes biological variation, instead of technical variation. As totalVI explicitly models nuisance factors (for genes as well as proteins), we can query the model while controlling for this nuisance variation. Furthermore, because naive computations of correlations on denoised values (parameters of conditional distributions) were shown to induce spurious gene-gene correlations [33], we develop a novel sampling scheme that helps remove technical variation while avoiding such artifacts.

In order to ensure our correlation matrix does not include variation from the modeled technical factors, we perturb the data generating process to fix the library size ($\ell = 10000$) as well as incorporate the denoised protein expression conditional distribution. In particular, we compute a correlation matrix using samples from the distribution

$$p(\log w_n, \log \tilde{r}_n | C_{1:N}, \ell_{1:N}). \tag{26}$$

This is also a posterior predictive density whose samples are generated with ancestral sampling. As $\tilde{r}_n$ is zero-inflated, we add the same "prior count" before taking the logarithm. For this distribution, we sample ancestrally using the aggregated posterior

$$q_\eta(z_n, \beta_n | C_{1:N}) = \frac{1}{N} \sum_{n=1}^{N} q_\eta(z_n | C_n) q_\eta(\beta_n | z_n, s_n), \tag{27}$$

One could in principle replace the aggregated posterior with the prior in case of analyzing dataset-wide correlations. However, this approach is more flexible as it can be applied to calculate the correlation matrix for a subpopulation $A = \{a_1, a_2, \ldots, a_m\}$, where A is the set of indices for the subpopulation, by conditioning the distribution on x_A and y_A.

The distinction between this procedure and those that induced spurious correlations is that the latter effectively estimates a correlation matrix using the expected value of the posterior predictive distribution, rather than estimating the correlation matrix of the posterior predictive distribution.

Out-of-batch generalization—totalVI learns a transformation from z_n and s_n to the parameters of the conditional distributions for each feature (decoder). In an out-of-batch prediction, we predict the expression of a cell (e.g., the mean of conditional distribution) given any of the other B observed batches s such that $s \neq s_n$. Here we describe a general way to sample posterior quantities for a cell while also "transforming" it into a different batch that was also observed for other cells [69]. Special cases of this have already been described in the protein imputation and differential expression sections. Consider, for instance, the RNA counts in cell n and gene g. We can calculate posterior predictive samples of $x_{n,g}$ while conditioning on any arbitrary observed batch b. Then,

$$p(x_{ng}^* | C_n, s = b) \approx \int p_v(x_{ng}^* | z_n, s = b) q_\eta(z_n | C_n) dz_n \,. \tag{28}$$

Furthermore, we can integrate over the choice of batch by sampling from

$$\sum_b p(x_{ng}^* | C_n, s = b) p(s = b), \tag{29}$$

where $p(s)$ is a uniform prior over batches. We take the expected value of this particular distribution as batch-corrected, denoised gene expression data. This "transforming" can also be applied to other likelihood parameters like π_n.

CITE-seq experiment on mouse spleen and lymph node

Supplementary Table 1 shows a summary of the experimental design that generated the mouse spleen and lymph node CITE-seq datasets. Below, we describe in further detail how these datasets were collected and processed.

Cell preparation—Mice were group housed with enrichment in standard cages on ventilated racks at an ambient temperature of 26C and 40% humidity. Mice were kept in a dark/light cycle of 12 hours on and 12 hours off. Two female C57BL/6 (B6) mice at 5 weeks

of age were euthanized using CO2. From each mouse, six lymph nodes were harvested, pooled in RPMI +10% FBS media on ice, mechanically dissociated with a syringe plunger, and passed through a 70 μm strainer to generate a single cell suspension. Likewise, the spleen was harvested, placed in RPMI +10% FBS media on ice, mechanically dissociated with a syringe plunger, and passed through a 70 μm strainer to generate a single cell suspension. For the spleen, red blood cells were lysed in Red Blood Cell Lysis Buffer (BioLegend # 420302) following the manufacturer's protocol. All animal care and procedures were carried out in accordance with guidelines approved by the Institutional Animal Care and Use Committee at BioLegend, Inc.

Antibody panel preparation—We prepared panels containing either 111 antibodies (TotalSeq-A mouse antibody panel 1, BioLegend # 900003217) or 208 antibodies (TotalSeq-A mouse antibody panel 2, BioLegend # 900003218), which are enumerated in Supplementary Data. We performed a buffer exchange on each panel using a 50kDa Amicon spin column (Millipore # UFC505096) following the manufacturer's protocol to transfer antibodies into RPMI + 10% FBS. Spleen and lymph node cell suspensions were stained with different hashtag antibodies [22].

CITE-seq protocol and library preparation—The CITE-seq experiment was performed following the TotalSeq protocol with two slight modifications. First, the 10 minute centrifugation at 14,000g to remove antibody aggregates was conducted prior to buffer exchange. Second, cells were stained, washed, and resuspended in RPMI + 10% FBS to maintain viability. After staining, washing, and counting, 12,000 spleen cells and 12,000 lymph node cells were mixed and loaded into a single 10x lane. We followed the 10x Genomics Chromium Single Cell 3' v3 protocol to prepare RNA, antibody-derived-tag (ADT) and hashtag-oligo (HTO) libraries [70].

Sequencing and data processing—RNA, ADT, and HTO libraries were sequenced with an Illumina NovaSeq S1. Reads were processed with Cell Ranger v3.1.0 with feature barcoding, where RNA reads were mapped to the mouse mm10–2.1.0 reference (10x Genomics, STAR aligner [71]) and antibody reads were mapped to known barcodes (Supplementary Table 5). Hashtags were demultiplexed separately for each 10x lane with HTODemux in Seurat v3 using the kmeans function [34]. No read depth normalization was applied when aggregating datasets.

Additional datasets

We also used publicly available CITE-seq datasets from 10x Genomics. These included "10k PBMCs from a Healthy Donor - Gene Expression and Cell Surface Protein" (PBMC10k, [23]), "5k Peripheral blood mononuclear cells (PBMCs) from a healthy donor with cell surface proteins (v3 chemistry)" (PBMC5k, [37]), and "10k Cells from a MALT Tumor - Gene Expression and Cell Surface Protein" (MALT, [24]). PBMC10k had 14,010 mean reads per cell for antibodies (5,816 median UMI counts per cell), while PBMC5k had 7,451 mean reads per cell for antibodies (2,752 median UMI counts per cell).

CITE-seq data pre-processing

For each dataset, after initial cell and gene filtering, we retained at least the top 4,000 highly variable genes (HVGs) as defined by the Seurat v3 method, merging HVGs from different batches when appropriate [34]. Dataset specific filtering is described below.

Spleen and lymph node—An initial cell filter removed cells expressing fewer than 200 genes. Cells labeled as either doublets or negative for hashtag antibodies by HTODemux were also removed. A protein library size filter retained cells with between 400 and 10,000 total protein UMI counts. We also filtered on the number of proteins detected. For cells stained with the 111 antibody panel, we removed cells with fewer than 90 proteins detected, while the cutoff was set to 170 for cells stained with the 208 antibody panel. Cells with a high percentage of UMIs from mitochondrial genes (15% or more of the cell's total UMI count) were removed. An initial gene filter removed genes expressed in 3 or fewer cells in any given batch. In addition to the top 4,000 HVGs selected by the Seurat v3 method, we retained genes that encode the proteins targeted by the 111 antibody panel. This resulted in 4,005 total genes. After all filters, the distribution of cells per dataset was: (SLN111-D1, 9,264 cells), (SLN111-D2, 7,564 cells), (SLN208-D1, 8,715 cells), (SLN208-D2, 7,105 cells). This is a total of 32,648 cells. Unless otherwise stated, we filtered out isotype control antibodies (9 total in the 208 panel) and hashtag antibodies. The protein CD49f was also removed due to having very low total UMI counts.

PBMC10k, PBMC5k, & MALT—For each of these datasets, we first removed doublets using DoubletDetection [72]. Cells with high mitochondrial content (percentage of UMIs from mitochondrial genes), high number of genes detected, high UMI counts, and with fewer than 200 genes expressed were removed. Next, cells with outlier protein library size (on either end) were removed. Genes with expression in three or fewer cells were removed. Finally, the top 4,000 HVGs were retained. Dataset specific parameters are in Supplementary Table 6. In the case where the PBMC datasets are integrated, the 4,000 HVGs are selected by merging HVGs computed on each dataset separately as in the Seurat v3 method.

Posterior predictive checks and held-out metrics

Posterior predictive checks are useful to check the fit of Bayesian models. They work by comparing the observed data to posterior predictive samples from the model [25]. Much of the benchmarking done here was inspired by previous work done to benchmark the scHPF model [16]. We compared totalVI to factor analysis, which is a linear-Gaussian alternative to totalVI, and is easily extendable to multiple modalities as features are treated conditionally independent of the latent representation. Furthermore, we compared to scHPF, which received the concatenated RNA and protein count matrices as input. As a control, we also compared performance on RNA only to scVI [15]. Posterior predictive samples for totalVI and scVI were obtained by calling the generate function in the scVI package, which samples from the variational posterior distributions, and subsequently from the likelihood distributions given the posterior samples. We ran scVI with 20 latent dimensions and negative binomial conditional distribution in order to be consistent with totalVI. Factor analysis (FA) models were fit using the sklearn package [73] on the combined RNA and

protein measurements using one of two normalization procedures. The first procedure consisted of transforming each value by log(count+1). The second procedure consisted of log library size normalizing the RNA features and protein features separately. For example, considering only the RNA measurements for a cell, we normalized each cell to sum to 1 by dividing by the library size of RNA, multiplied by 10,000, added 1 to each value, and took a log transformation:

$$\tilde{x}_{ng} = \log\left(L\frac{x_{ng}}{\sum_g x_{ng}} + 1\right), \tag{30}$$

where $L = 10000$. This process was then applied to the protein measurements. We refer to this type of normalization as *log library size normalization*, and for short, *log rate*. These normalization procedures are necessary as FA assumes a Gaussian distribution, so training on the raw data would lead to poor model fit. Posterior predictive samples for FA models were computed using the fitted parameters and posterior distribution derived in [74]. We note that normalization procedures were inverted so that FA posterior predictive samples were on the same scale as the raw data.

For each dataset, each model was trained on a train set comprising of 85% of the cells. An additional 5% of cells were held-out as a validation set for totalVI early stopping. The remaining 10% of cells were also held-out as a test set. For each model's posterior predictive samples (25 for each model) based on the train set, we calculated the coefficient of variation (CV) for each feature, and calculated the mean absolute error between the average CV and the observed raw data CV. Furthermore, we computed the Mann-Whitney U statistic (implementation in `scipy.stats.mannwhitneyu`) for each feature between the posterior predictive sample and the raw data. We averaged the statistic across all posterior predictive samples for each feature. We also used posterior predictive samples to assess generalization to unseen data. In this setup, we generated posterior predictive samples (150 for each model) conditioned on the test set. We considered the mean absolute error between the observed held-out data and the posterior predictive mean.

Moreover, we computed a held-out calibration error [26] for each model based on the test set. For each cell n and gene g, let I_{ng} be the indicator that the observed value is contained in the interval of all posterior predictive samples. The calibration error for genes is then calculated as

$$\text{Cal}^{\text{RNA}} = \left(1 - \frac{1}{NG}\sum_n \sum_g I_{ng}\right)^2. \tag{31}$$

The calibration error for proteins is computed separately following the same procedure.

Finally, for totalVI and scVI only, and for only the RNA data, we computed the held-out predictive log likelihood. In this metric, z_n and ℓ_n were sampled from the variational posterior for each cell n and the average negative conditional log likelihood, $-\log p(x_n | z_n, \ell_n, s_n)$ was computed. This is also called the reconstruction loss in the VAE

literature. This is also an approximation of $-\log p(x_n | x_n, y_n, s_n)$, the negative predictive log likelihood. We note that we cannot compare the log likelihood of totalVI and scVI, which use discrete conditional distributions to factor analysis models, which use continuous conditional distributions.

We further evaluated model misfit through posterior dispersion indices [75]. This metric highlights cells that are not well explained by the model. This analysis is described in Supplementary Note 7.

Background decoupling benchmarking

We reported the totalVI background probability as the posterior predictive mean of π_{nt}, thus

$$p(\text{cell } n, \text{ protein } t \text{ is background}) = \mathbb{E}_{p(\pi_{nt} | x_n, y_n, s_n)}[\pi_{nt}], \tag{32}$$

where the expectation is approximated using Monte Carlo integration. The totalVI foreground probability is one minus the background probability.

Observing protein background in empty droplets and non-expressing cell types—To observe different sources of protein background, we considered both empty droplets and cell types with known expression of surface markers. We defined empty droplets as non-cell barcodes from the SLN111-D1 dataset with between 20 and 100 RNA UMI counts (approximately 75,358 barcodes). We chose these criteria so that empty droplets were likely to represent ambient molecules rather than sequencing errors (with very low UMI counts) or cell debris (with higher UMI counts) [76]. To observe non-specific binding of antibodies, we considered B cells (which are known to express CD19 and CD20, but not CD28) and T cells (which are known to express CD28, but not CD19 or CD20). Using cell type annotations as described below, we grouped all high-quality, non-doublet B cell clusters (excluding plasma B cells), and alpha/beta T cell clusters (including all CD4, Treg, and CD8 T cell clusters). We observed that for these three proteins, both empty droplets and the non-expressing cell type contained protein background (non-zero protein counts) with varying degrees of overlap with the foreground signal of the expressing cell type. In this text, we describe the protein counts of the non-expressing cell type above the counts in empty droplets as non-specific antibody binding, although we acknowledge there could be multiple sources of this cell-specific background (Supplementary Note 3).

Classification of cell type by marker proteins—We sought to evaluate totalVI against a Gaussian mixture model (GMM) at predicting major cell types by the foreground probability of commonly used surface markers. For these markers, protein counts were expected to come from the foreground component in some cell types and from the background component in others. For example, a high foreground probability for CD4 could be used as a positive predictor of CD4 T cells. We applied scikit-learn's `GaussianMixture` with default parameters to fit a GMM with two components to the $\log(\text{protein counts} + 1)$ for each protein for all cells in the SLN111-D1 dataset. We interpreted the posterior probability of the component with the higher mean as the foreground probability and that of the lower mean as the background probability. Restricting all cells to just those that fell into the

categories of B cells or T cells as described above, we tested how well totalVI or a GMM could classify cell types based on commonly used protein surface markers. For each marker protein, we computed a receiver operating characteristic curve (ROC) (`sklearn.metrics`) by thresholding the totalVI or GMM foreground probability estimates, using manual cell type annotations as true labels (stratification and annotation described below). We reported the area under the ROC (ROC AUC). The cell type considered as the positive population was either B cells, T cells, CD4 T cells, or CD8 T cells depending on the marker. In tests considering each of these positive populations, all remaining cells among the B and T cells were considered the negative population. Marker proteins tested included, for B cells: CD19, CD45R-B220, CD20, I-A-I-E (MHC II); for T cells: CD5, TCRb, CD28, CD90.2; for CD4 T cells: CD4; for CD8T cells: CD8a, CD8b [77–80]. Although we are aware of documented exceptions to these markers appearing strictly on a single cell type (e.g., CD5 is expressed on a portion of B1 B cells), these exceptions are rare. In these cases where marker expression is not mutually exclusive, cell types can still be distinguished by the gradation in levels of the marker between cell populations. Thus, these exceptions do not negate the utility of these markers in broad cell type classification (which is apparent in both totalVI and GMM performance at this classification task).

GMM-based cutoff for protein foreground/background—As a baseline determination of a cutoff to distinguish cells with foreground or background protein expression, we used a GMM fit on all cells of the SLN111-D1 dataset for each protein as described above. The GMM-based cutoff between foreground and background was determined to be the protein expression level at which the GMM foreground probability (described above) was closest to 0.5.

Protein normalization using isotype controls—Although totalVI does not make use of isotype controls in its model of protein background, some CITE-seq studies include isotype control antibodies as negative controls to adjust for protein background. To compare totalVI to a method that uses isotype controls to normalize protein data, we applied two different normalization strategies prior to fitting a GMM and performing the classification task described above. First, we applied the normalization strategy used by Cumulus [32]:

$$\text{norm1}: y_{nt} \rightarrow \max\left(\log\frac{y_{nt}+1}{k_n^{(t)}+1}, 0\right), \tag{33}$$

where y_{nt} is the observed UMI counts for protein t in cell n, and $k_n^{(t)}$ is the observed UMI counts of the corresponding isotype control for protein t in cell n. In the case where the corresponding isotype control for a given antibody is not present in the data, normalized expression is calculated as

$$\text{norm1}: y_{nt} \rightarrow \log(y_{nt}+1)\,. \tag{34}$$

Because this normalization method restricts normalized values to be non-negative, the resulting distribution might not be fit well by a GMM. We therefore applied a second normalization strategy as a modification to the Cumulus method that adjusts for the relative

isotype control level but does not restrict the distribution of normalized values to be non-negative:

$$\text{norm2}: y_{nt} \rightarrow \log \frac{y_{nt} + 1}{k_n^{(t)} + 1}. \quad (35)$$

If an isotype control is not present, *norm2* values are calculated as in Equation 34.

For the SLN208-D1 dataset, which contained a limited number of isotype control antibodies, we fit a GMM as described above to the log(protein counts + 1) *(GMM)*, to the Cumulus normalized values *(GMM norm1)*, and to the values normalized with the modified Cumulus method *(GMM norm2)*. We performed the same classification of cell types by marker proteins as described for the SLN111-D1 dataset, noting that the isotype control for CD28 (Syrian Hamster IgG) was not contained in the dataset.

Visualization and raw data normalization

For the SLN111-D1 dataset, we visualized the totalVI latent space in two dimensions using Scanpy's [56] implementation of the UMAP algorithm [81]. We applied log library-size normalization to the raw RNA counts as in Equation 30. All cells of the SLN111-D1 dataset are plotted (i.e., doublets were not removed).

Distribution of foreground probabilities—We observed the totalVI foreground probability for all proteins across all cells in the SLN111-D1 dataset (Supplementary Fig. 1e). The totalVI foreground probability tended to fall near zero or one. Measurements for which totalVI estimates a foreground probability near 0.5 are instances where the model is uncertain about whether the measurement is likely to be derived from foreground or background.

Distinguishing foreground and background in trimodal protein distributions—Despite using a two-component mixture, totalVI can decouple the background and foreground of proteins that have more than two modes globally. totalVI is capable of distinguishing foreground and background in this setting because the mixture is conditionally dependent on z_n, which allows the foreground and background expression modes to be defined locally in the latent space. For example, as has been reported using flow cytometry [82], CITE-seq data of peripheral blood mononuclear cells contains three modes of CD4 expression corresponding to CD4 T cells, monocytes, and background. totalVI detected that both CD4 T cells and monocytes had foreground expression of CD4, while the CD4 expression of the remaining cells was attributable to background expression.

Denoised protein expression—Denoised protein expression was calculated as previously described. B cells and T cells were defined by annotations, as described above.

RNA-protein correlation analysis

Evaluation of correlation calculation with permuted features—Using totalVI, we aimed to calculate a correlation matrix between all RNA and protein features free from

nuisance variation such as sequencing depth and protein background. We took care to avoid the naive calculation of correlations directly between denoised features, noting that a recent study reported false positive correlations in smoothed scRNA-seq data [33]. Instead, we developed a novel sampling method for the calculation of denoised feature correlations that removes nuisance variation while avoiding imputation-induced artifacts (described above).

To evaluate whether totalVI could calculate a denoised feature correlation matrix without introducing spurious relationships in the data, we permuted the expression of a set of genes to serve as a negative control. To create this set of negative control genes from the SLN111-D1 dataset, we selected the 100 genes with highest mean expression that were not already among the top highly variable genes used in the model. We randomly permuted the counts of these genes within each cell, rendering these genes independent of all other gene and protein features. After concatenating the SLN111-D1 dataset with the permuted gene expression for all cells, we ran the totalVI model.

We then calculated Pearson and Spearman correlations between features using three methods, referred to here as raw, naive totalVI, and totalVI correlations. Raw correlations were calculated between log library-size normalized RNA (Equation 30) and log(protein counts + 1). Naive totalVI correlations were calculated between totalVI denoised RNA and totalVI denoised proteins. totalVI correlations were calculated by sampling denoised RNA and denoised protein values from the posterior (as described above).

We observed the correlations between all RNA and protein features as well as the 100 additional genes whose expression was randomly permuted. By comparing the raw correlations with denoised correlations, we observed whether the method of denoising could maintain the relationship between these permuted genes and other features, which, in expectation, are independent from each other. Here, we highlighted the correlations between all proteins and the randomly permuted genes, whose correlations are expected to be near zero.

Correlations of RNA-protein pairs—We calculated a feature correlation matrix for the SLN111-D1 dataset using either the totalVI sampling method or by calculating raw correlations as described above. The resulting feature correlation matrices for both Pearson and Spearman correlations were subset to each protein and its encoding RNA for all proteins with a unique encoding RNA in the dataset (i.e., excluding RNA with multiple isoforms such as Ptprc). It is worth noting that the totalVI model has no explicit information about the relationship between RNA-protein pairs, such that any correlation learned by the model is not predetermined by known RNA-protein relationships.

Integration of multiple datasets

We compared totalVI's integration performance to that of Scanorama[35], Seurat v3 [34], and Harmony [36]. The two former methods, like totalVI, produce both integrated expression values and integrated low-dimensional cell representations. The input to both Scanorama (`scanorama.correct`), Seurat v3 (`FindIntegrationAnchors`, `IntegrateData`) methods was a normalized matrix of concatenated genes and proteins. Genes were subset to the same subset used as input to totalVI. The RNA counts of this

matrix were normalized following standard log library size normalization (Equation 30). For proteins, we used a $y \rightarrow \log(y+1)$ transformation. Finally, we standard scaled each feature. Harmony (`harmonypy`) received latent spaces for each dataset computed with PCA on the concatenated, normalized, and scaled datasets. All methods were run with default parameters. We compared the performance of the methods using the following metrics:

Latent mixing metric—The latent mixing metric measures how well the latent cell representations are mixed between batches relative to the global frequency of batches. First, a cell-cell similarity matrix is computed from a latent representation of cells. Next, select 100 cells uniformly at random, and calculate the frequency of batches represented in each cell's 100 nearest neighbors. Let $p_i^{(n)}$ be the frequency of batch i in the 100 nearest neighbors of cell n. Let q_i be the global frequency of batch i. Compute the negative relative entropy between the frequency of observed batches in the neighborhood, and the global frequency of batches:

$$\mathrm{KL}\left(p^{(n)} \parallel q\right) = \sum_{i=1}^{B} p_i^{(n)} \log \frac{p_i^{(n)}}{q_i} \tag{36}$$

Repeat this 50 times and return the average negative relative entropy. This is conceptually similar to the entropy of mixing that has been used in other studies [83].

Measurement mixing metric—The measurement mixing metric describes how well the high-dimensional measurements are batch corrected, and for each feature, is related to the Mann-Whitney U statistic. Consider one feature in the batch-corrected data matrix placed in rank order. Let R_1 be the sum of the ranks of the cells in batch 1 and N_1 be the number of cells in batch 1. Define $U_1 = R_1 - \frac{N_1(N_1+1)}{2}$. Similarly, compute U_2 for batch 2 and return $\min(U_1, U_2)$. Higher values of this metric indicate better mixing within that feature. This metric could not be applied for Harmony, which only produces an integrated latent representation.

Feature retention metric—The feature retention metric describes how spatial autocorrelation of both RNA and protein change when comparing cells from an integrated latent representation to a latent representation derived from each batch separately. Lower values of this metric indicate that the integration procedure reduced the localization of feature expression, indicating some degree of random mixing. We calculate it as follows. For two batches and a particular integration method, we calculate Z_1 and Z_2, the latent representations of the cells of batch 1 and batch 2, respectively. The latent space computation of the individual batches was chosen to closely match the integration method (see below). We also calculate an integrated latent representation of both batches $Z^\top = \left[\widetilde{Z}_1 \ \widetilde{Z}_2\right]$. Let $D_1 = [X_1 \ Y_1]$ be the combined RNA and protein batch 1 in which RNA is library size log normalized and proteins are log-transformed. Let $\mathbb{E}[H(D_1, Z_1)]$ be the expected feature autocorrelation score as calculated by Hotspot [84]. Furthermore, let $\mathbb{E}\left[H\left(D_1, \widetilde{Z}_1\right)\right]$ be the

analogous quantity calculated using the latent cell representations of batch 1 subsetted from the joint, integrated representation. The feature retention metric is calculated as $\frac{1}{2}\sum_i^2 \mathbb{E}\left[H\left(D_i, \widetilde{Z}_i\right)\right] - \mathbb{E}[H(D_i, Z_i)]$. In the case of totalVI union, features were intersected to compute this metric.

For Scanorama, we define Z_1 and Z_2 to be a 100-dimensional matrix produced with principal components analysis (PCA), which is the same dimension reduction used in the integration method. For Seurat v3, we similarly use PCA to reduce D_1 and D_2 to 30 dimensions, the same number of dimensions used for integration. The input to PCA was the same as the input for the respective method, except for Scanorama, where we additionally L_2 normalized each cell, because this step is done automatically by Scanorama's correct method.

Clustering metric—The clustering metric quantifies the extent to which clusters defined on the unintegrated latent spaces are preserved in the integrated latent space. Using the same notation as before, we compute for each method, clusters based on Z_1 and Z_2, individually. Clusters were inferred using the standard Scanpy workflow: computing a neighbors graph, and running the Leiden [85] algorithm, with default parameters. Next, the silhouette coefficient $\mathcal{S}$ was computed for every cell with respect to its latent representation and cluster label:$\mathcal{S}(Z_1)$, $\mathcal{S}(Z_2)$, $\mathcal{S}\left(\widetilde{Z}_1\right)$,$\mathcal{S}\left(\widetilde{Z}_2\right)$. Finally, a score for each dataset was defined as $\mathbb{E}\left[\mathcal{S}\left(\widetilde{Z}_i\right) - \mathcal{S}(Z_i)\right]$. The final score was averaged across each dataset. Thus, lower scores suggest clusters are not preserved as well in the integrated latent space. We emphasize that this metric can only be taken as a proxy for cell type preservation, which requires "ground truth" cell type labels, or well-established datasets -- none of which exist for CITE-seq.

Missing protein imputation—For Seurat v3, we imputed proteins based on mutual nearest neighbors in the RNA data using the FindTransferAnchors and TransferData functions. Again, RNA data were log library size normalized. Proteins were not normalized as input to Seurat. For totalVI, after fitting the model, cells from the batch with held-out proteins were decoded conditioned on being in the batch with observed protein data. Note, we did not correct for background in this analysis since the comparison is to the observed data. We used the root mean squared error of values on the log scale to assess imputation accuracy. To produce error bars, we ran totalVI 30 times, resampling the dataset into the train/validation sets (validation used for early stopping), computing the mean and 95% confidence interval. For the PBMC datasets, we compared to cTP-net [38], which is a neural network that was pre-trained on specific CITE-seq datasets from human cells, with no option to train a new dataset. The inputs to cTP-net were the log-normalized RNA data. cTP-net did not provide predictions for CD127, CD15, CD25, PD-1, or TIGIT. To the best of our knowledge, neither of the PBMC datasets used in this study were used to train the pre-trained cTP-net model. Thus, a direct comparison of the results to those of totalVI or Seurat v3 is not straightforward.

Stratification of cells in SLN-all

We stratified cells of the mouse spleen and lymph node based on the SLN-all dataset (totalVI-intersect model fit as described above). We clustered cells in the totalVI latent space with Scanpy's implementation of the Leiden algorithm at resolution 1, resulting in 32 clusters [56, 85]. We repeated this approach to sub-cluster cells, finding a total of 43 clusters. We used Vision [49] with default parameters for data exploration, including its implementation of the Wilcoxon rank sum test, to identify cluster markers. Clusters were manually annotated based on a curated list of cell type markers (Supplementary Table 4). Clusters annotated as doublets, low quality cells (e.g., high percentage of UMI counts from mitochondrial genes), or cells undergoing the cell cycle were removed from further analysis. Again, we visualized the totalVI latent space in two dimensions using Scanpy's implementation of the UMAP algorithm. These annotations were also consistent with the latent space derived with totalVI-union (Supplementary Fig. 7).

Differential expression analysis

The Welch's t-test and Wilcoxon rank-sum test for each differential expression scenario were run on protein features (log-transformed) using the Scanpy library, which produces adjusted p-values corrected for multiple testing by the Benjamini-Hochberg procedure [86]. Both tests are two-sided. A protein was considered to be differentially expressed if the adjusted p-value was less than 0.05. Each application of totalVI differential expression tests to a dataset requires a trained totalVI model. For each dataset used in DE analysis, all cells were included to train the model. Throughout, we used our manual annotations from the SLN-all totalVI-intersect model run. The cells in nuisance clusters (described in previous section) were removed before running totalVI differential expression functions.

In a given totalVI differential expression test, we identified cell type markers by first filtering features for significance (log Bayes factor > 0.7), and then sorting by the median log fold change. We only retained genes with non-zero UMI counts in at least 10% of the subset of cells.

In the comparison to scVI gene Bayes factors, each method was trained independently on the SLN111-D1 dataset. We ran scVI with 20 latent dimensions and negative binomial conditional distribution to be consistent with totalVI. Differential expression of genes in scVI was computed using the same LFC-based method, which is implemented in the scvi-tools package. In reproducibility benchmarking, totalVI was trained independently on the replicates.

In the test between ICOS-high Tregs and CD4 conventional T cells, we used the same totalVI-intersect model fit that was used to manually annotate the cells. In this test, we expected CD73, CD357 (GITR), CD122, and CD5 to be upregulated (positives) in ICOS-high Tregs relative to conventional CD4 T cells [87–90]. The list of putative negatives included I-A/I-E (MHC II), IgD, CD19, CD8b, and CD8a, which have no expected expression in either of these cell types.

DE on imputed proteins—In one totalVI model fit, SLN111-D1 and SLN111-D2 were integrated with the proteins of SLN111-D2 held out. In the second totalVI model fit, these two datasets were integrated with all data. In testing differential expression of proteins, and for each model fit, we conditioned on SLN111-D1. This is an application of Equation 22, except that the prior $p(s')$ is 1 if $s' = \text{SLN111-D1}$ and 0 otherwise.

Archetypal analysis

This analysis was performed on the SLN-all totalVI-intersect model run. As z_n is distributed as logistic normal, the latent space is then constrained to the probability simplex (i.e., each z_n is non-negative and sums to one). Archetypes correspond to vertices of the totalVI latent space, which means they can be represented by the identity matrix I_d, where d is the number of latent dimensions (20 in all experiments). In this setting, the latent space is the 19-dimensional standard simplex.

We first identified and removed four archetypes from further interpretation that suffered from inactivity (a known issue in training VAEs) [91]. For the remaining 16 latent dimensions, we decoded the archetypes to obtain denoised RNA and protein archetypal expression profiles, all conditioned on batch 0 (the SLN111-D1 experiment). We then computed denoised RNA and protein expression profiles for all cells in SLN-all, conditioned on SLN111-D1. To derive signatures for each archetype, we computed the mean and standard deviation of each feature in the denoised RNA and protein expression matrices (without the archetypes) and standard scaled the archetypal profiles with respect to this mean and standard deviation. We refer to this quantity as the archetype score. The top features for each archetype were those with an archetype score greater than 2. The distance to the archetype is computed as the Manhattan distance from each cell's latent representation to the archetype. The distances per archeytpe were scaled into the range $[0, 1]$.

B cell analysis

For this analysis, we used the totalVI-intersect model fit on the SLN-all dataset as described above. The SLN-all dataset was filtered to include all high-quality, non-doublet clusters annotated as B cells (excluding plasma B cells), resulting in 15,560 cells.

Calculation of signature scores—Gene signature analysis was conducted using Vision [49] with default parameters. Gene signatures, including interferon response signatures, were downloaded from MSigDB gene sets [92]. Signature scores were calculated on all cells in the SLN-all dataset based on cell similarities defined by t latent space.

Identification of transitional and mature B cell feature modules—totalVI Spearman correlations between all features were calculated separately within the transitional B cell cluster and the mature B cell cluster. Features were subset by the following method. From a one-vs-one DE test between transitional and mature B cells, we selected the top ten marker genes and top three marker proteins for each cluster (as described above). We added to this list the four features most highly correlated with each differentially expressed feature within its respective cluster. This resulted in a list of both transitional and mature features which we used to subset the full feature correlation matrix. Features were hierarchically

clustered separately for transitional and mature B cells using Seaborn's clustermap with default parameters.

When plotting totalVI expression of each feature as a function of $1 - Z_{16}$, each feature was standard scaled and smoothed with a loess curve. Spearman correlations were calculated between each feature and $1 - Z_{16}$. The p-values of these correlations were all significant (BH-adjusted p-value < 0.001).

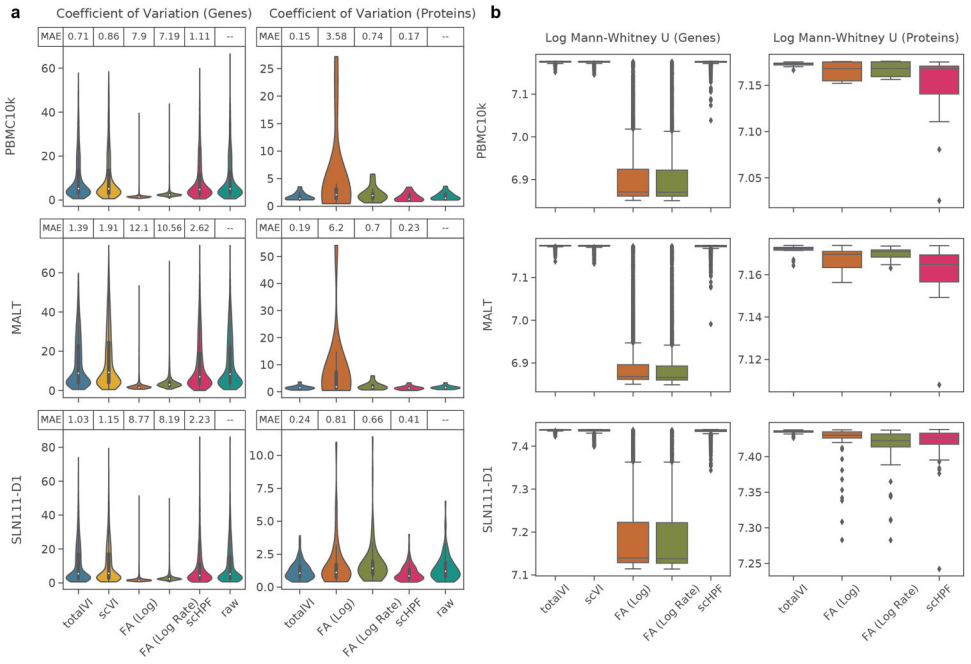
Data availability

The data discussed in this manuscript (SLN-all) have been deposited in NCBI's Gene Expression Omnibus [93] and are accessible through GEO Series accession number GSE150599 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE150599). Processed data are also available in the reproducibility GitHub repository (https://github.com/YosefLab/totalVI_reproducibility). The SLN-all dataset processed with totalVI can be explored interactively with Vision at http://s133.cs.berkeley.edu:9000/Results.html. Public datasets were downloaded from 10x Genomics (PBMC5k: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.2/5k_pbmc_protein_v3; PBMC10k: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_10k_protein_v3; MALT: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/malt_10k_protein_v3). Mouse mm10 reference was downloaded from 10x Genomics

Code availability

The code to reproduce the results in this manuscript is available at https://github.com/YosefLab/totalVI_reproducibility and has been deposited at https://doi.org/10.5281/zenodo.4330368 [94]. The reference implementation of totalVI is available via the scvi-tools package at https://github.com/YosefLab/scvi-tools

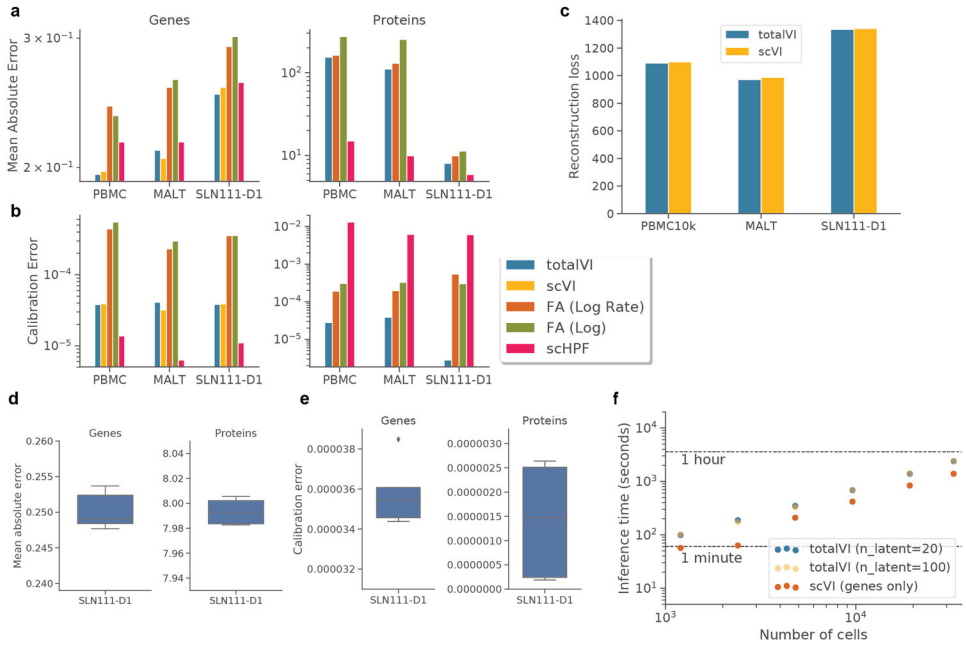
Extended Data



Extended Data Fig. 1.

Evaluation of totalVI model.

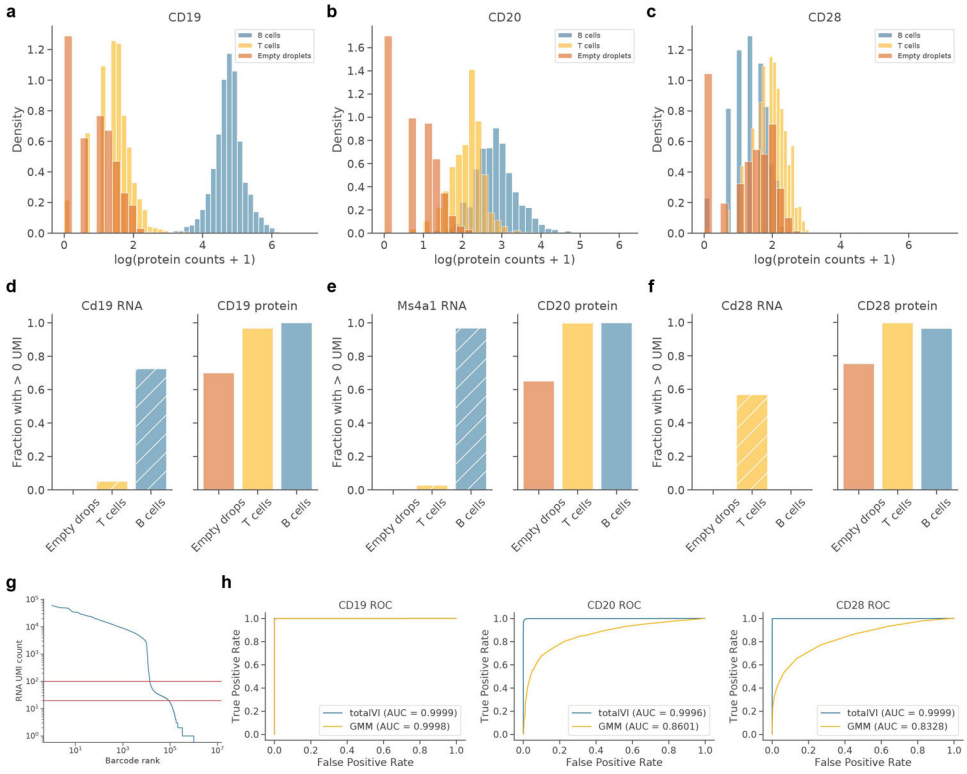
a, Posterior predictive check of coefficient of variation (CV) of genes and proteins. For each of the PBMC10k, MALT, and SLN111-D1 datasets and for each model (totalVI, scVI, factor analysis with normalized input, scHPF) the average coefficient of variation from posterior predictive samples was computed for each feature. Violin plots summarize the distribution of CVs for genes and proteins. Mean absolute error (MAE) between raw data CVs and average posterior predictive CV are reported. **b**, For each gene and protein, the Mann-Whitney U statistic between posterior predictive samples and observed data averaged over samples. Shown are boxplots of this statistic for each set of features (genes and proteins), model, and dataset (n=4000 genes across datasets and n=14 proteins for PBMC10k and MALT, n=110 proteins for SLN111-D1). Box plots indicate the median (center line), interquartile range (hinges), and whiskers at 1.5x interquartile range. Higher is better.



Extended Data Fig. 2.

Evaluation of totalVI model (continued).

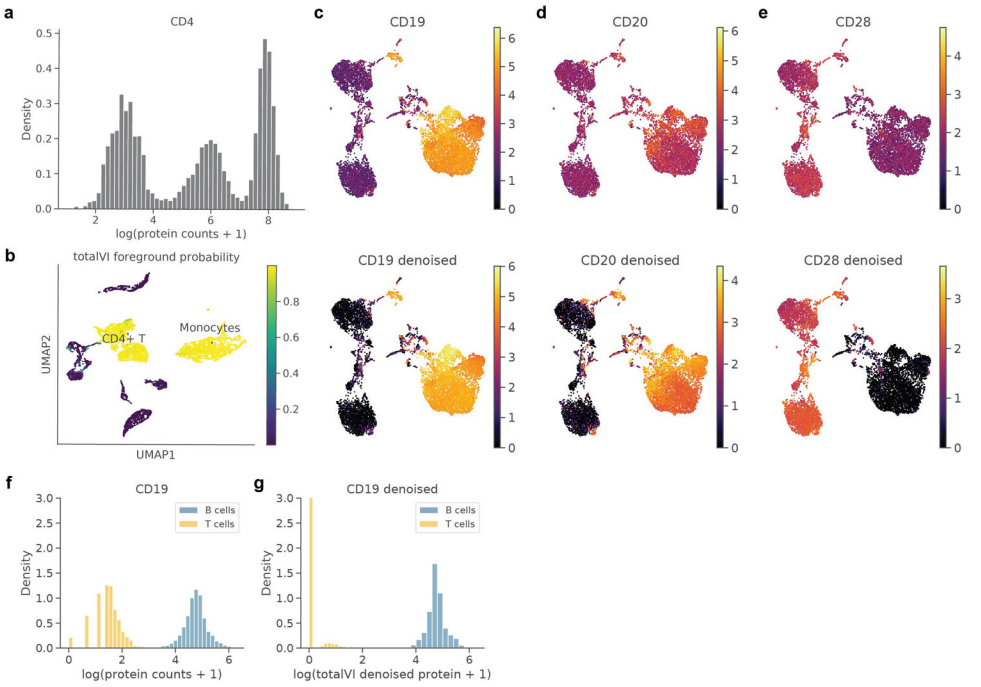
a, Mean absolute error (MAE) between held out data and posterior predictive mean separated by genes and proteins for each model and dataset. **b**, Calibration error of held-out data reported separately for genes and proteins. **c**, Held-out reconstruction loss of RNA for scVI and totalVI. **d, e**, Stability of held-out results (n=5 initializations) for totalVI on SLN111-D1. Metrics displayed are the **(d)** Held out MAE, and **(e)** held out calibration error. Box plots indicate the median (center line), interquartile range (hinges), and whiskers at 1.5x interquartile range. **f**, Inference time for totalVI and scVI across cells randomly subsampled to different levels from SLN-all. scVI was run with only genes. totalVI was applied with 20 latent dimensions and 100 latent dimensions.



Extended Data Fig. 3.

Protein background in cells and empty droplets

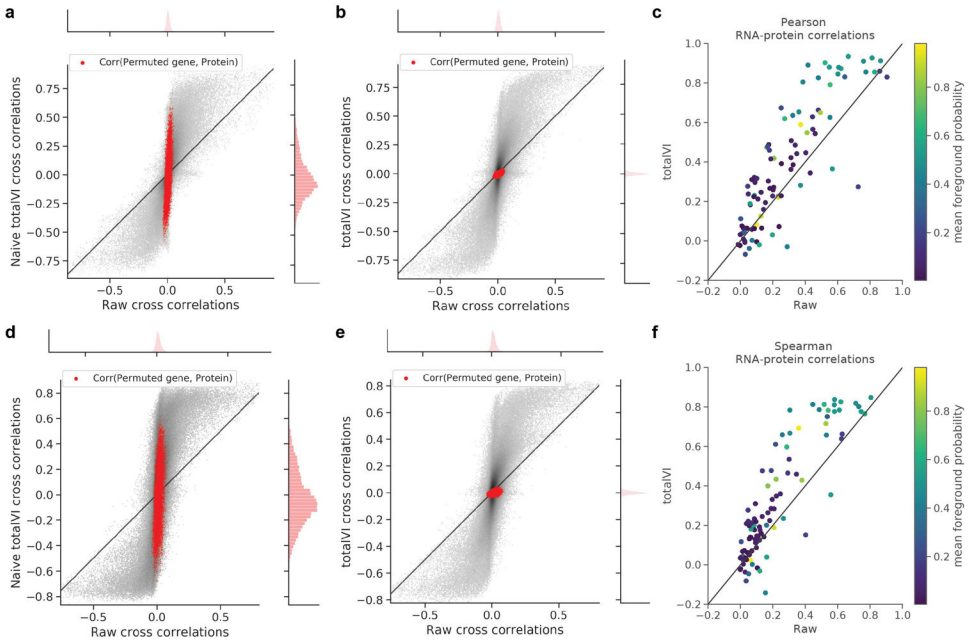
a–c, Histogram of log(protein counts + 1) in the SLN111-D1 dataset for B cells, T cells, and empty droplets (Methods) for CD19 **(a)**, CD20 **(b)**, and CD28 **(c)**. d-f, Fraction of empty droplets, B cells, or T cells with > 0 UMIs detected for a given RNA (left, hatched) or protein (right, solid). RNA/proteins displayed are Cd19/CD19 **(d)**, Ms4a1/CD20 **(e)**, and Cd28/CD28 **(f)**. **g**, Barcode rank plot for all barcodes detected in the SLN111-D1 dataset. Red lines at 20 and 100 RNA UMI counts indicate the lower and upper bounds, respectively, used to define empty droplets in **(a–f)**. **h**, Performance of totalVI and a Gaussian mixture model (GMM) fit on all cells for each protein of the SLN111-D1 dataset to classify cell types by marker proteins (Methods). Receiver operating characteristic (ROC) curves shown for CD19 (B cells), CD20 (B cells), or CD28 (T cells). Area under the receiver operating characteristic curve (ROC AUC score) was calculated using as input either the totalVI foreground probability or GMM foreground probability where the indicated cell type was the positive population out of all B and T cells.



Extended Data Fig. 4.

totalVI decouples foreground and background for trimodal protein distributions and denoises protein data

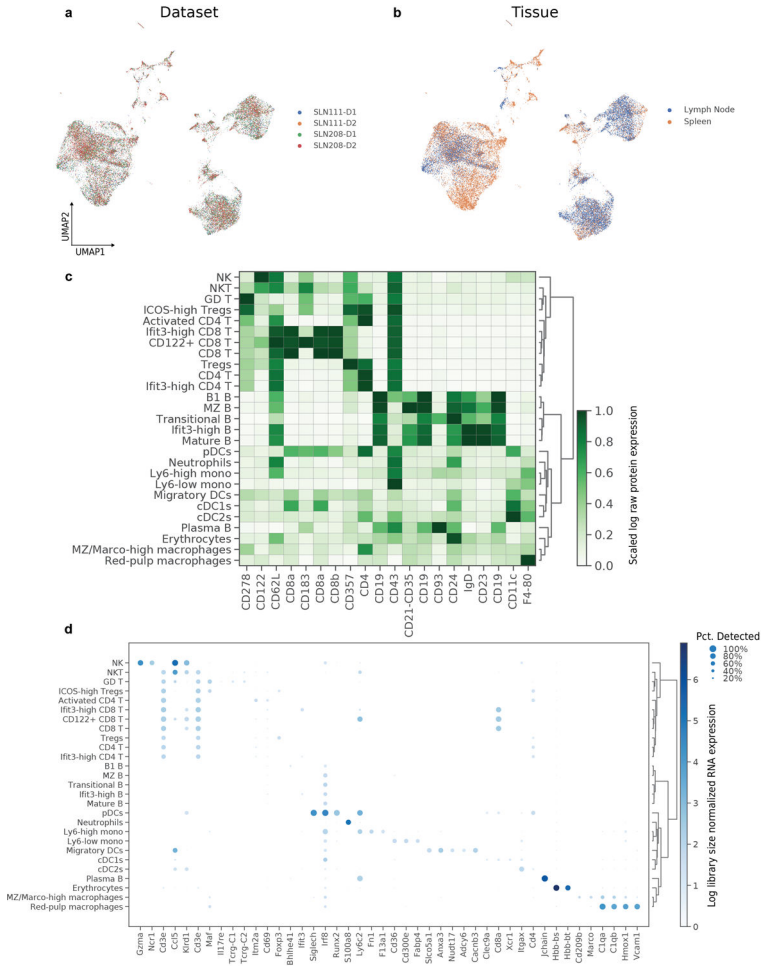
a, b, CD4 protein expression in the PBMC10k dataset. **(a)** Trimodal distribution of log(protein counts + 1). **(b)** UMAP plot of the totalVI latent space colored by totalVI foreground probability. **c-e**, UMAP plots of the totalVI latent space for the SLN111-D1 dataset. Plots are colored by log(protein counts+1) (top) and log(totalVI denoised protein+1) (bottom) for CD19 **(c)**, CD20 **(d)**, and CD28 (e). **f, g**, Distributions of log(protein counts + 1) **(f)** and log(totalVI denoised protein + 1) **(g)** for CD19 protein in B and T cells. y-axis is truncated at 3.



Extended Data Fig. 5.

RNA-protein correlations

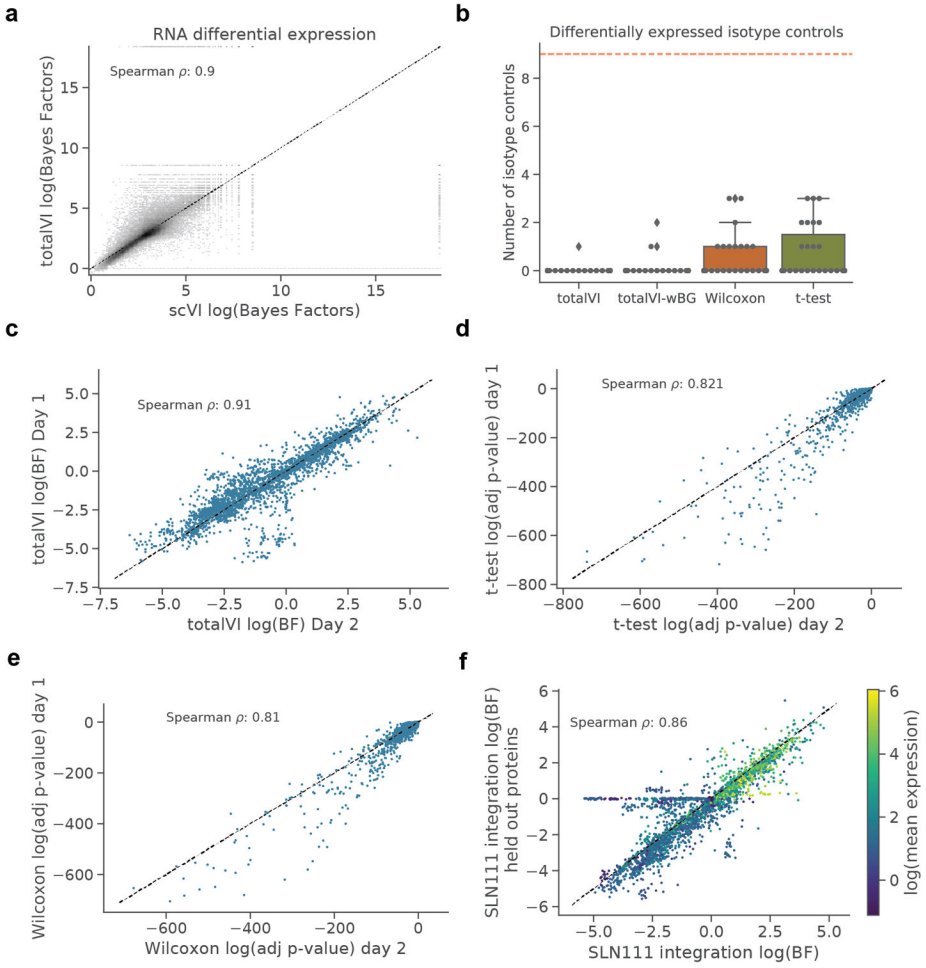
a, b, 2d density plots of Pearson correlations between all RNA and protein features in the SLN111-D1 dataset as well as 100 additional genes whose expression was randomly permuted. Correlations between all proteins and the randomly permuted genes are colored in red. Raw correlations were calculated between log library-size normalized RNA and log(protein counts + 1). **(a)**, Naive totalVI correlations were calculated between totalVI denoised RNA and totalVI denoised proteins. **(b)**, totalVI correlations were calculated between denoised RNA and proteins sampled from the posterior (Methods). **c**, Pearson correlations between each protein and its encoding RNA for all proteins with a unique encoding RNA, colored by the mean probability foreground of the protein across all cells. totalVI correlations were calculated as in (b) and raw correlation were calculated as in **(a, b)**. **d-f**, Same as **(a-c)**, but for Spearman correlations.



Extended Data Fig. 6.

Integration of SLN-all with totalVI-intersect

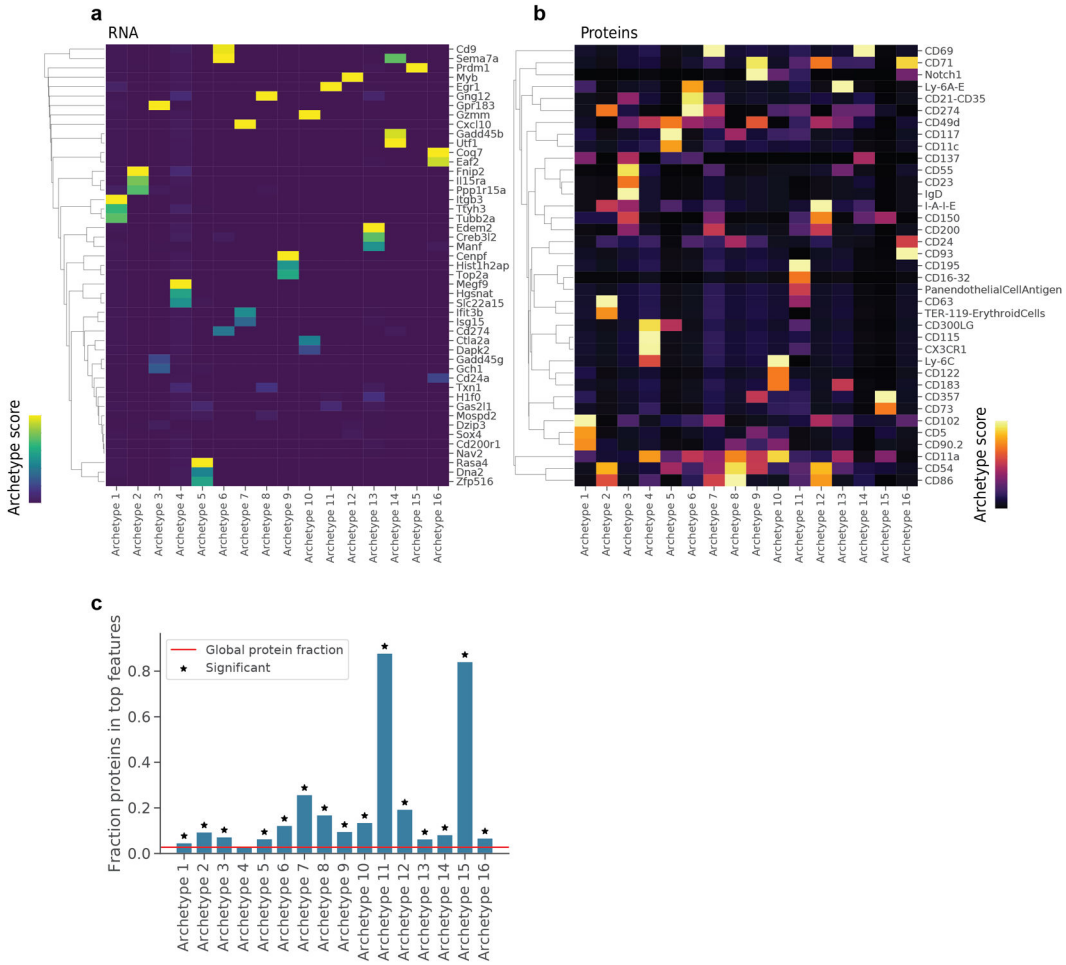
a, b, UMAP plot of SLN-all colored by **(a)** dataset, and **(b)** tissue. **c**, Heatmap of proteins used for annotation. Proteins (columns) are log(protein counts + 1) scaled by column for visualization. **d**, Dotplot of RNA markers used for annotation. RNA is log library size normalized.



Extended Data Fig. 7.

Differential expression analysis

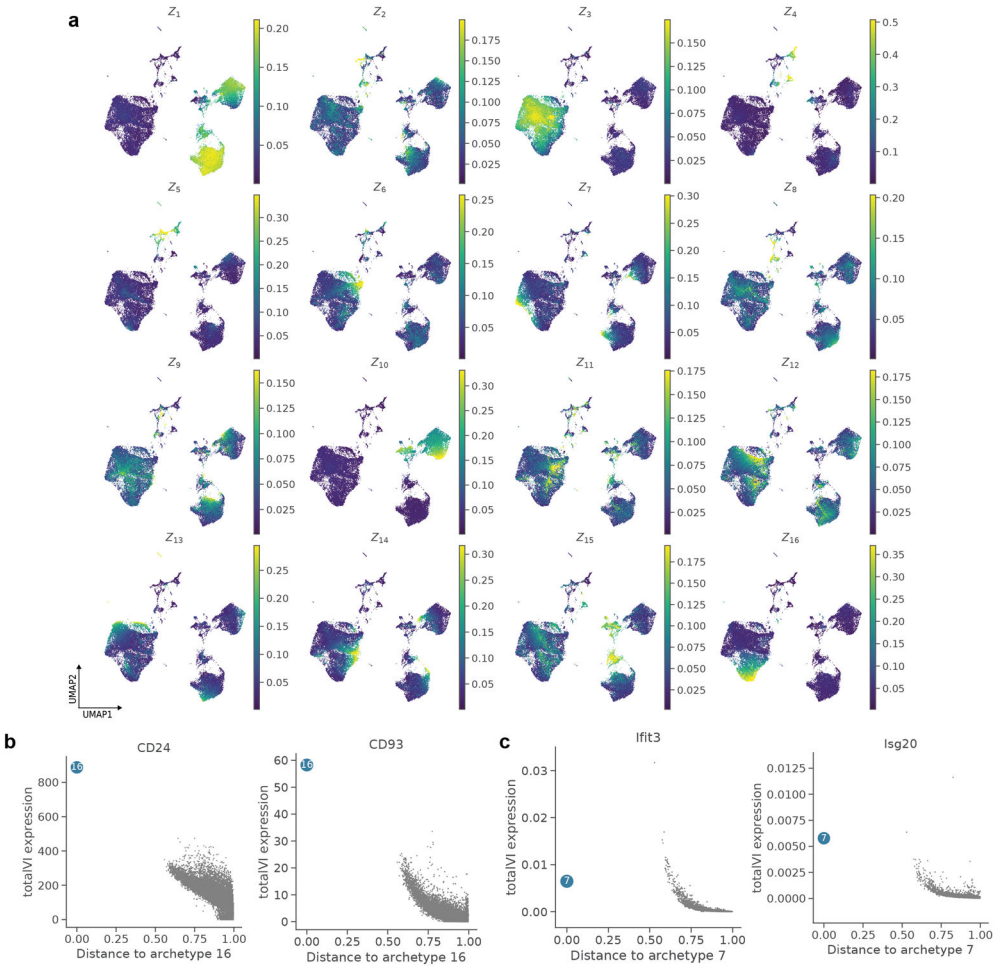
a, 2d density plot of totalVI and scVI log Bayes factors for genes. Bayes factors were computed for each gene in one-vs-all tests on the SLN111-D1 dataset. **b**, Number of isotype controls called differentially expressed in one-vs-all tests (n=27) for totalVI, totalVI-wBG (totalVI test without background removal), Wilcoxon rank-sum, and t-test. Tests were applied to SLN208-D1, for which isotype controls were retained. Box plots indicate the median (center lines), interquartile range (hinges), whiskers at 1.5x interquartile range. Red dashed line indicates the maximum number of isotype controls. **c-e**, Significance level (Bayes factors for totalVI, adjusted p-value for frequentist tests) for proteins in one-vs-all tests computed on SLN111-D1 and SLN111-D2 for each of **(c)** totalVI, **(d)** t-test, **(e)** Wilcoxon. **f**, Bayes factors for proteins in one-vs-all tests computed on the SLN111 datasets integrated with and without the SLN111-D2 proteins held-out. Differential expression tests for both model fits were conditioned on SLN111-D1. Bayes factors are colored by the average protein expression from SLN111-D1.



Extended Data Fig. 8.

Interpreting totalVI latent dimensions with archetypal analysis.

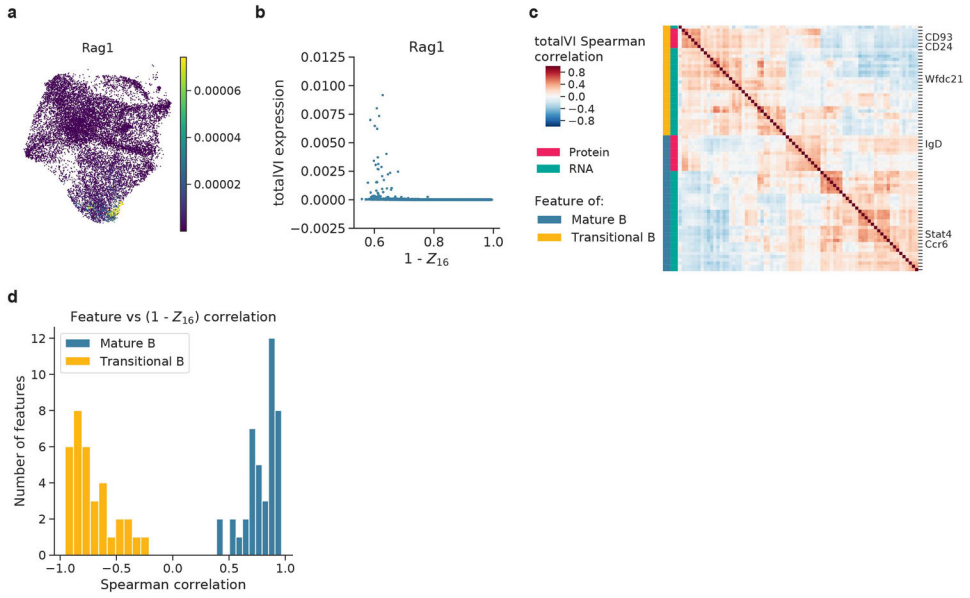
a, b, Heatmap of top **(a)** gene and **(b)** protein features for each archetype. The archetype score corresponds to the standard scaled archetypal expression profiles (Methods). Heatmaps are individually column normalized for visualization. **c**, Fraction of proteins in top archetypal features for each archetype. Features in each archetype were selected in the “top” if they had an archetype score of greater than 2. For these features, we performed a one-sided hypergeometric test to determine if proteins were over-represented in this feature set relative to the global distribution of feature types. Archetypes with over-representation of proteins (one-sided hypergeometric test, BH-adjusted p-value <0.05) are denoted.



Extended Data Fig. 9.

Visualization of archetypes in totalVI-intersect model of SLN-all

a, UMAP plots of SLN-all cells colored by latent dimension value. **b**, totalVI protein expression for CD24 and CD93 proteins as a function of distance to archetype 16. **c**, totalVI denoised expression for Isg20 and Ifit3 genes as a function of distance to archetype 7. Archetype is colored in blue, all other cells in grey.



Extended Data Fig. 10.

totalVI identifies correlated modules of RNA and proteins that are associated with the maturation of transitional B cells

a, totalVI Spearman correlations in mature B cells between the same RNA and proteins as in Figure 5h. Features were hierarchically clustered within mature B cells. **b**, UMAP of the totalVI latent space colored by totalVI RNA expression of Rag1. **c**, totalVI RNA expression of Rag1 as a function of $1 - Z_{16}$ (the totalVI latent dimension associated with transitional B cells). **d**, Histogram of Spearman correlations between each feature in **(a)** and $1 - Z_{16}$ (n = 2,735 cells).

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

We thank Ellen Robey, Lydia Lutes, and Derek Bangs for help designing experiments. We thank BioLegend Inc. and their proteogenomics team, especially Bertrand Yeung, Andre Fernandes, Qing Gao, Hong Zhang, Tse Shun Huang, for providing reagents and expertise and for help with sample preparation, library generation, and sequencing of CITE-seq libraries. We thank David DeTomaso for general data analysis advice, and Pierre Boyeau, Achille Nazaret, and Galen Xing for help with integrating totalVI in the scvi-tools package. We thank members of the Streets and Yosef laboratories for helpful feedback. Research reported in this manuscript was supported by the NIGMS of the National Institutes of Health under award number R35GM124916 (A.S), the Chan-Zuckerberg Foundation Network under grant number 2019–02452 (N.Y.), and the National Institutes of Mental Health under grant number U19MH114821 (N.Y.). A.G. is supported by NIH Training Grant 5T32HG000047–19. Z.S. is supported by the National Science Foundation Graduate Research Fellowship. N.Y. was supported by the Koret-Berkeley-Tel Aviv (KBT) Initiative in Computational Biology. A.S. and N.Y. are Chan Zuckerberg Biohub investigators.

References

1. Stubbington MJT, Rozenblatt-Rosen O, Regev A & Teichmann SA Single-cell transcriptomics to explore the immune system in health and disease. Science 358, 58–63 (2017). [PubMed: 28983043]

2. Papalexi E & Satija R Single-cell RNA sequencing to explore immune cell heterogeneity. Nature Reviews Immunology (2017) 10.1038/nri.2017.76.

3. Labib M & Kelley SO Single-cell analysis targeting the proteome. Nature Reviews Chemistry 4, 143–158 (2020).

4. Wagner A, Regev A & Yosef N Revealing the vectors of cellular identity with single-cell genomics. Nature Biotechnology (2016) 10.1038/nbt.3711.

5. Efremova M & Tiechmann SA Computational methods for single-cell omics across modalities. Nature Methods (2020).

6. Stoeckius M et al. Simultaneous epitope and transcriptome measurement in single cells. Nature Methods (2017) 10.1038/nmeth.4380.

7. Peterson VM et al. Multiplexed quantification of proteins and transcripts in single cells. Nature Biotechnology (2017) 10.1038/nbt.3973.

8. Regev A et al. The Human Cell Atlas. eLife (2017) 10.7554/eLife.27041.

9. Tanay A & Regev A Scaling single-cell genomics from phenomenology to mechanism. Nature (2017) 10.1038/nature21350.

10. Todorovic V Single-cell RNA-seq—now with protein. Nature Methods 14, 1028–1029 (2017).

11. Haque A, Engel J, Teichmann SA & Lönnberg T A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. Genome Medicine 9, 1–12 (2017). [PubMed: 28081715]

12. Granja JM et al. Single-cell multiomic analysis identifies regulatory programs in mixed-phenotype acute leukemia. Nature Biotechnology 37, 1458–1465 (2019).

13. Praktiknjo SD et al. Tracing tumorigenesis in a solid tumor model at single-cell resolution. Nature Communications 11, 991 (2020).

14. Kotliarov Y et al. Broad immune activation underlies shared set point signatures for vaccine responsiveness in healthy individuals and disease activity in patients with lupus. Nature Medicine 26, 618–629 (2020).

15. Lopez R, Regier J, Cole MB, Jordan MI & Yosef N Deep generative modeling for single-cell transcriptomics. Nature Methods 15, 1053–1058 (2018). [PubMed: 30504886]

16. Levitin HM et al. De novo gene signature identification from single-cell RNA -seq with hierarchical Poisson factorization. Molecular Systems Biology 15, (2019).

17. Azizi E, Prabhakaran S, Carr A & Pe'er D Bayesian inference for single-cell clustering and imputing. Genomics and Computational Biology (2017) 10.18547/gcb.2017.vol3.iss1.e46.

18. Risso D, Perraudeau F, Gribkova S, Dudoit S & Vert JP A general and flexible method for signal extraction from single-cell RNA-seq data. Nature Communications (2018) 10.1038/s41467-017-02554-5.

19. Blei DM Build, compute, critique, repeat: Data analysis with latent variable models. Annual Review of Statistics and Its Application (2014) 10.1146/annurev-statistics-022513-115657.

20. Kingma DP & Welling M Auto-Encoding variational Bayes. in International conference on learning representations (2014).

21. Cutler A & Breiman L Archetypal analysis. Technometrics (1994) 10.1080/00401706.1994.10485840.

22. Stoeckius M et al. Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. Genome Biology (2018) 10.1186/s13059-018-1603-1.

23. 10X Genomics. 10k PBMCs from a Healthy Donor - gene expression and cell surface protein. (2018).

24. 10X Genomics. 10k Cells from a MALT Tumor - gene expression and cell surface protein. (2018).

25. Gelman A, Meng XL & Stern H Posterior predictive assessment of model fitness via realized discrepancies. Statistica Sinica (1996).

26. Kuleshov V, Fenner N & Ermon S Accurate uncertainties for deep learning using calibrated regression. in International conference on machine learning (2018).

27. Hulspas R, O'Gorman MRG, Wood BL, Gratama JW & Sutherland DR Considerations for the control of background fluorescence in clinical flow cytometry. Cytometry Part B: Clinical Cytometry (2009) 10.1002/cyto.b.20485.

28. Yang S et al. Decontamination of ambient RNA in single-cell RNA-seq with DecontX. Genome Biology 21, 57 (2020). [PubMed: 32138770]
29. Young MD & Behjati S SoupX removes ambient RNA contamination from droplet based single cell RNA sequencing data. bioRxiv 303727 (2018) 10.1101/303727.
30. Fleming SJ, Marioni JC & Babadi M CellBender remove-background: a deep generative model for unsupervised removal of background noise from scRNA-seq datasets. bioRxiv (2019) 10.1101/791699.
31. Ngo Trong T et al. Semisupervised generative autoencoder for single-cell data. Journal of Computational Biology (2019) 10.1089/cmb.2019.0337.
32. Li B et al. Cumulus provides cloud-based data analysis for large-scale single-cell and single-nucleus RNA-seq. Nature Methods 17, 793–798 (2020). [PubMed: 32719530]
33. Andrews TS & Hemberg M False signals induced by single-cell imputation. F1000Research (2019) 10.12688/f1000research.16613.2.
34. Stuart T et al. Comprehensive Integration of Single-Cell Data. Cell 177, 1888–1902 (2019). [PubMed: 31178118]
35. Hie B, Bryson B & Berger B Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nature Biotechnology (2019) 10.1038/s41587-019-0113-3.
36. Korsunsky I et al. Fast, sensitive and accurate integration of single-cell data with Harmony. Nature Methods 16, (2019).
37. 10X Genomics. 5k Peripheral blood mononuclear cells (PBMCs) from a healthy donor with cell surface proteins (v3 chemistry). (2019).
38. Zhou Z, Ye C, Wang J & Zhang NR Surface protein imputation from single cell transcriptomes by deep neural networks. Nature Communications 11, 1–10 (2020).
39. Kass RE & Raftery AE Bayes factors. Journal of the American Statistical Association 90, 773–795 (1995).
40. Boyeau P et al. Deep Generative Models for Detecting Differential Expression in Single Cells. in Machine learning in computational biology (2019). 10.1101/794289.
41. Bezman NA et al. Molecular definition of the identity and activation of natural killer cells. Nature Immunology 13, 1000–1008 (2012). [PubMed: 22902830]
42. Walzer T et al. Identification, activation, and selective in vivo ablation of mouse NK cells via NKp46. Proceedings of the National Academy of Sciences of the United States of America 104, 3384–3389 (2007). [PubMed: 17360655]
43. Gordon SM et al. The transcription factors T-bet and Eomes control key checkpoints of natural killer cell maturation. Immunity 36, 55–67 (2012). [PubMed: 22261438]
44. Korem Y et al. Geometry of the gene expression space of individual cells. PLoS Computational Biology 11, 1–27 (2015).
45. Dijk D van et al. Finding archetypal spaces for data using neural networks. arXiv (2019).
46. Thomas MD, Srivastava B & Allman D Regulation of peripheral B cell maturation. Cellular Immunology 239, 92–102 (2006). [PubMed: 16797504]
47. Loder F et al. B cell development in the spleen takes place in discrete steps and is determined by the quality of B cell receptor-derived signals. Journal of Experimental Medicine 190, 75–89 (1999).
48. Kreslavsky T et al. Essential role for the transcription factor Bhlhe41 in regulating the development, self-renewal and BCR repertoire of B-1a cells. Nature Immunology 18, 442–455 (2017). [PubMed: 28250425]
49. DeTomaso D et al. Functional interpretation of single cell similarity maps. Nature Communications (2019).
50. Lock EF, Hoadley KA, Marron JS & Nobel AB Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. Annals of Applied Statistics 7, 523–542 (2013).
51. Argelaguet R et al. Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. Molecular Systems Biology 14, 1–13 (2018).
52. Liu Y, Beyer A & Aebersold R On the dependency of cellular protein levels on mRNA abundance. Cell 165, 535–550 (2016). [PubMed: 27104977]

53. Gorin G, Svensson V & Pachter L Protein velocity and acceleration from single-cell multiomics experiments. Genome Biology 21, 1–6 (2020).
54. Svensson V, Beltrame E. da V. & Pachter L Quantifying the tradeoff between sequencing depth and cell number in single-cell RNA-seq. bioRxiv (2019) 10.1101/762773.
55. Heimberg G, Bhatnagar R, El-Samad H & Thomson M Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. Cell Systems 2, 239–250 (2016). [PubMed: 27135536]
56. Wolf FA, Angerer P & Theis FJ SCANPY: Large-scale single-cell gene expression data analysis. Genome Biology (2018) 10.1186/s13059-017-1382-0.
57. Clark SJ et al. ScNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells e. Nature Communications 9, 1–9 (2018).
58. Mimitou EP et al. Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. Nature Methods 16, 409–412 (2019). [PubMed: 31011186]
59. Svensson V, Gayoso A, Yosef N & Pachter L Interpretable factor models of single-cell RNA-seq via variational autoencoders. Bioinformatics (2020) 10.1101/737601.
60. Wang C & Blei DM A general method for robust Bayesian modeling. Bayesian Analysis (2018) 10.1214/17-BA1090.
61. Svensson V Droplet scRNA-seq is not zero-inflated. Nature Biotechnology (2020).
62. Blei DM, Kucukelbir A & McAuliffe JD Variational Inference: A review for statisticians. Journal of the American Statistical Association 112, 859–877 (2017).
63. Sønderby CK, Raiko T, Maaløe L, Sønderby SK & Winther O Ladder variational autoencoders. in Neural information processing systems (2016).
64. Kingma DP & Ba JL Adam: A method for stochastic optimization. in International conference on learning representations (2015).
65. Lopez R et al. A joint model of unpaired data from scRNA-seq and spatial transcriptomics for imputing missing gene expression measurements. in ICML workshop in computational biology (2019).
66. Mattei PA & Freiisen J Miwae: Deep generative modelling and imputation of incomplete data sets. in International conference on machine learning (2019).
67. Blitzer J, Crammer K, Kulesza A, Pereira F & Wortman J Learning bounds for domain adaptation. in Advances in neural information processing systems (2008).
68. Ganin Y et al. Domain-adversarial training of neural networks. The Journal of Machine Learning Research 17, 2096–2030 (2016).
69. Lotfollahi M, Naghipourfar M, Theis FJ & Wolf FA Conditional out-of-sample generation for unpaired data using trVAE. arXiv (2019).
70. Zheng GXY et al. Massively parallel digital transcriptional profiling of single cells. Nature Communications (2017) 10.1038/ncomms14049.
71. Dobin A et al. STAR: Ultrafast universal RNA-seq aligner. Bioinformatics (2013) 10.1093/bioinformatics/bts635.
72. Gayoso Adam, Shor Jonathan, Carr Ambrose J., Sharma Roshan, Pe'er Dana (2018, 7 17). DoubletDetection (Version v2.4). Zenodo. doi: 10.5281/zenodo.2678041
73. Pedregosa F et al. Scikit-learn: machine learning in Python. Journal of Machine Learning Research (2011).
74. Bishop CM Pattern Recogniton and Machine Learning. (2006).
75. Kucukelbir A, Wang Y & Blei DM Evaluating Bayesian models with posterior dispersion indices. in International Conference on Machine Learning (2017).
76. Lun ATL et al. EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. Genome Biology 20, 63 (2019). [PubMed: 30902100]
77. Lai L, Alaverdi N, Maltais L & Morse HC Immunophenotyping mouse cell surface antigens: Nomenclature and immunophenotyping. The Journal of Immunology (1998).
78. Watts C Capture and processing of exogenous antigens for presentation on MHC molecules. Annual Review of Immunology 15, 821–850 (1997).

79. Uchida J et al. Mouse CD20 expression and function. International Immunology (2004) 10.1093/intimm/dxh009.
80. Hünig T, Beyersdorf N & Kerkau T CD28 co-stimulation in T-cell homeostasis: a recent perspective. ImmunoTargets and Therapy 4, 111 (2015). [PubMed: 27471717]
81. McInnes L, Healy J & Melville J UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv (2018).
82. Filion LG, Izaguirre CA, Garber GE, Huebsh L & Aye MT Detection of surface and cytoplasmic CD4 on blood monocytes from normal and HIV-1 infected individuals. Journal of Immunological Methods 135, 59–69 (1990). [PubMed: 1703191]
83. Haghverdi L, Lun ATL, Morgan MD & Marioni JC Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nature Biotechnology 36, 421–427 (2018).
84. DeTomaso D & Yosef N Identifying informative gene modules across modalities of single cell genomics. bioRxiv (2020) 10.1101/2020.02.06.937805.
85. Traag V, Waltman L & Eck NJ van. From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports 9, (2019).
86. Benjamini Y & Hochberg Y Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B (Methodological) 57, 289–300 (1995).
87. Zhao H, Liao X & Kang Y Tregs: Where we are and what comes next? Frontiers in Immunology (2017) 10.3389/fimmu.2017.01578.
88. Roncarolo M-G & Gregori S Is FOXP3 a bona fide marker for human regulatory T cells? European Journal of Immunology 38, 925–927 (2008). [PubMed: 18395862]
89. Fontenot JD, Rasmussen JP, Gavin MA & Rudensky AY A function for interleukin 2 in Foxp3-expressing regulatory T cells. Nature Immunology 6, 1142–1151 (2005). [PubMed: 16227984]
90. Sprouse ML et al. High self-reactivity drives T-bet and potentiates Treg function in tissue-specific autoimmunity. JCI Insight 3, 1–14 (2018).
91. Burda Y, Grosse R & Salakhutdinov R Importance weighted Autoencoders. in International conference on learning representations (2016).
92. Liberzon A et al. Databases and ontologies Molecular signatures database (MSigDB) 3.0. Bioinformatics 27, 1739–1740 (2011). [PubMed: 21546393]
93. Edgar R, Domrachev M & Lash AE Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Research 30, 207–210 (2002). [PubMed: 11752295]
94. Gayoso A and Steier Z (2020, 12 18). YosefLab/totalVI_reproducibility: totalVI reproducibility v0.3 (Version v0.3). Zenodo. 10.5281/zenodo.4330368.

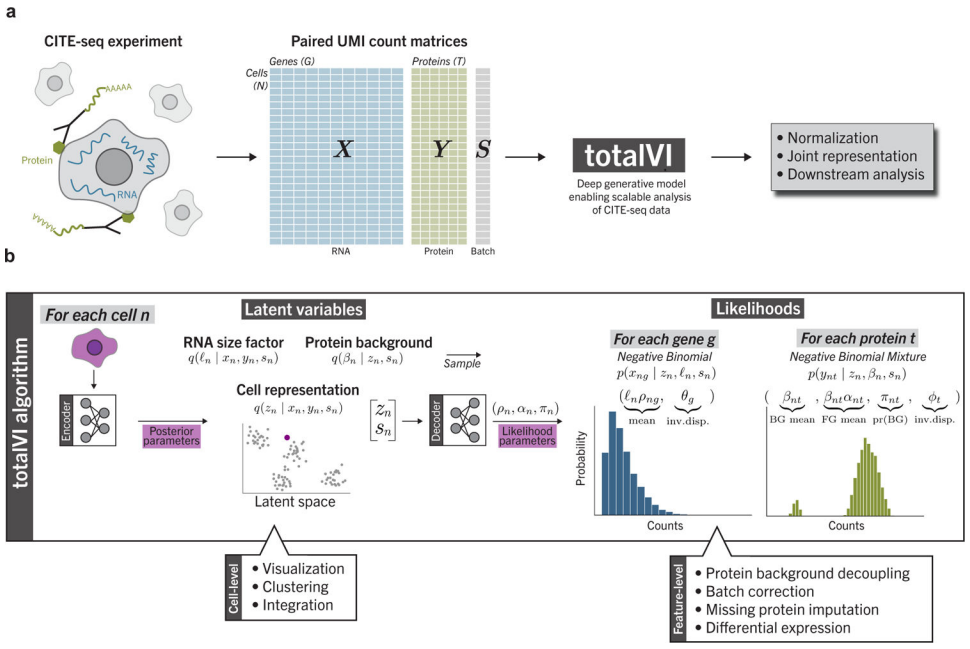


Figure 1: Schematic of a CITE-seq data analysis pipeline with totalVI.

a, A CITE-seq experiment simultaneously measures RNA and surface proteins molecules in single cells, producing paired count matrices for RNA and proteins. These matrices, along with an optional matrix containing sample-level categorical covariates (batch), are the input to totalVI, which concomitantly normalizes the data and learns a joint representation of the data that is suitable for downstream analysis tasks. **b**, Schematic of totalVI model. The RNA counts, protein counts, and batch for each cell n are jointly transformed by an encoder neural network into the parameters of the posterior distributions for z_n, a low-dimensional representation of cell state, β_n, the protein background mean indexed by protein, and ℓ_n, an RNA size factor. The posterior mean of z_n, which we refer to as the latent representation, is corrected for batch effects and can be used as input to clustering and visualization algorithms. Next, a decoder neural network maps samples from the posterior distribution of z_n, along with the batch, s_n, to parameters of a negative binomial distribution for each gene and a negative binomial mixture for each protein, which contains a foreground (FG) and background (BG) component (Methods). These parameters are used for feature-level analyses.

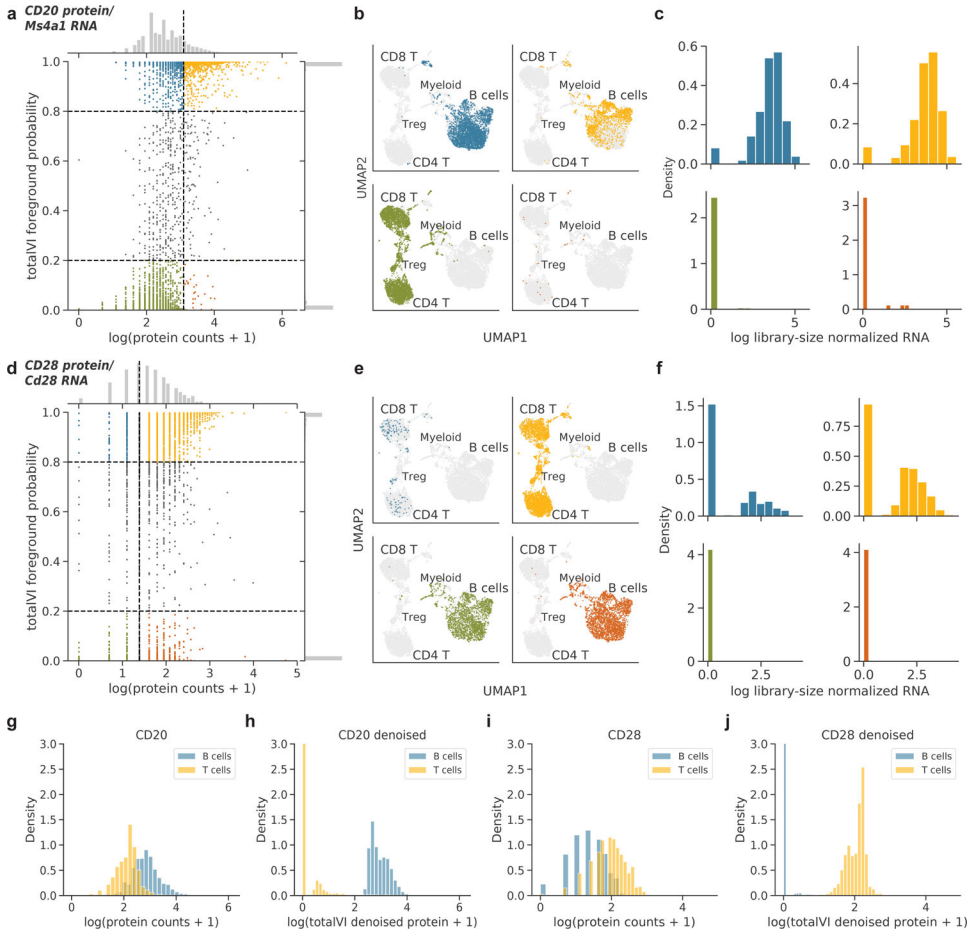


Figure 2: totalVI identifies and corrects for protein background.

totalVI was applied to the SLN111-D1 dataset. **a-c**, CD20 protein (encoded by Ms4a1 RNA). **(a)** totalVI foreground probability vs log(*protein counts* + 1). Vertical line denotes protein foreground/background cutoff determined by a GMM. Horizontal lines denote totalVI foreground probability of 0.2 and 0.8. Cells with foreground probability greater than 0.8 or less than 0.2 are colored by quadrant, while the remaining cells are gray. **(b)** UMAP plots of the totalVI latent space. Each quadrant contains cells from the corresponding quadrant of (a) in color with the remaining cells in gray. **(c)** RNA expression (log library-size normalized; Methods 4.8) for cells colored in (a). **d-f**, Same as (a-c), but for CD28 protein (encoded by Cd28 RNA). **g, h**, Distributions of log(*protein counts* + 1) **(g)** and log(*totalVI denoised protein* + 1) **(h)** for CD20 protein in B cells (blue) and T cells (yellow). y-axis is truncated at 3. **i, j**, Same as (g, h), but for CD28 protein.

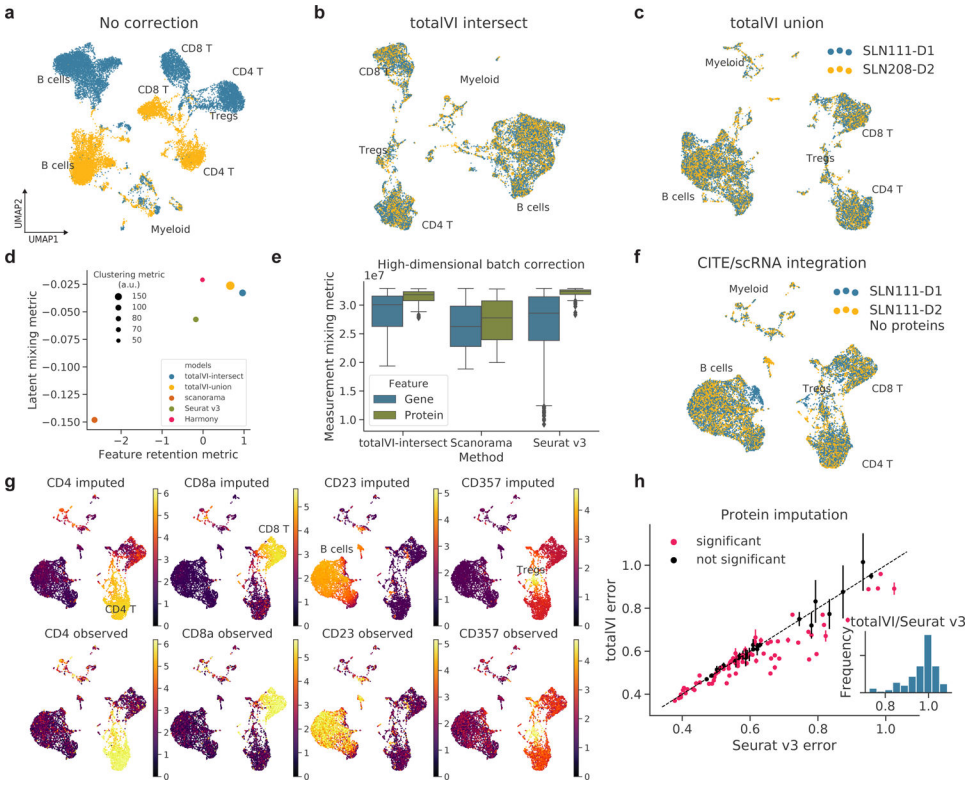


Figure 3: Benchmarking of integration methods for CITE-seq data.

a-c, UMAP plots of SLN111-D1 and SLN208-D2 with no integration (PCA of paired data with intersection of protein panels), and after integration with totalVI-intersect, in which the protein panels were intersected, and totalVI-union, in which the unequal protein panels were preserved, colored by dataset. **d, e**, Performance of integration methods based on four metrics: **(d)** latent mixing metric, feature retention metric, clustering metric (displayed as point size), and **(e)** measurement mixing metric (computed for $n = 4000$ genes and $n = 111$ proteins; higher values are better for each; Methods). Box plots indicate the median (center lines), interquartile range (hinges), whiskers at 1.5x interquartile range. **f**, UMAP plot of SLN111-D1 integrated with SLN111-D2 (proteins held out) by totalVI. **g**, UMAP plots colored by totalVI imputed and observed protein expression (log scale) of key cell type markers (range 0–99th percentile of held-out values for each protein). **h**, Root mean squared error (RMSLE) of imputed versus observed protein expression (log scale) for totalVI-union and Seurat v3. totalVI performance per protein is presented as mean RMSLE with error bars representing 95% confidence intervals of the mean estimate ($n = 30$ model initializations). Proteins colored in black are not significantly different in performance, while those in red are significantly different (two-sided Student's t-test, BH-adjusted p-value < 0.05). Inset displays ratio in performance across proteins for totalVI and Seurat v3.

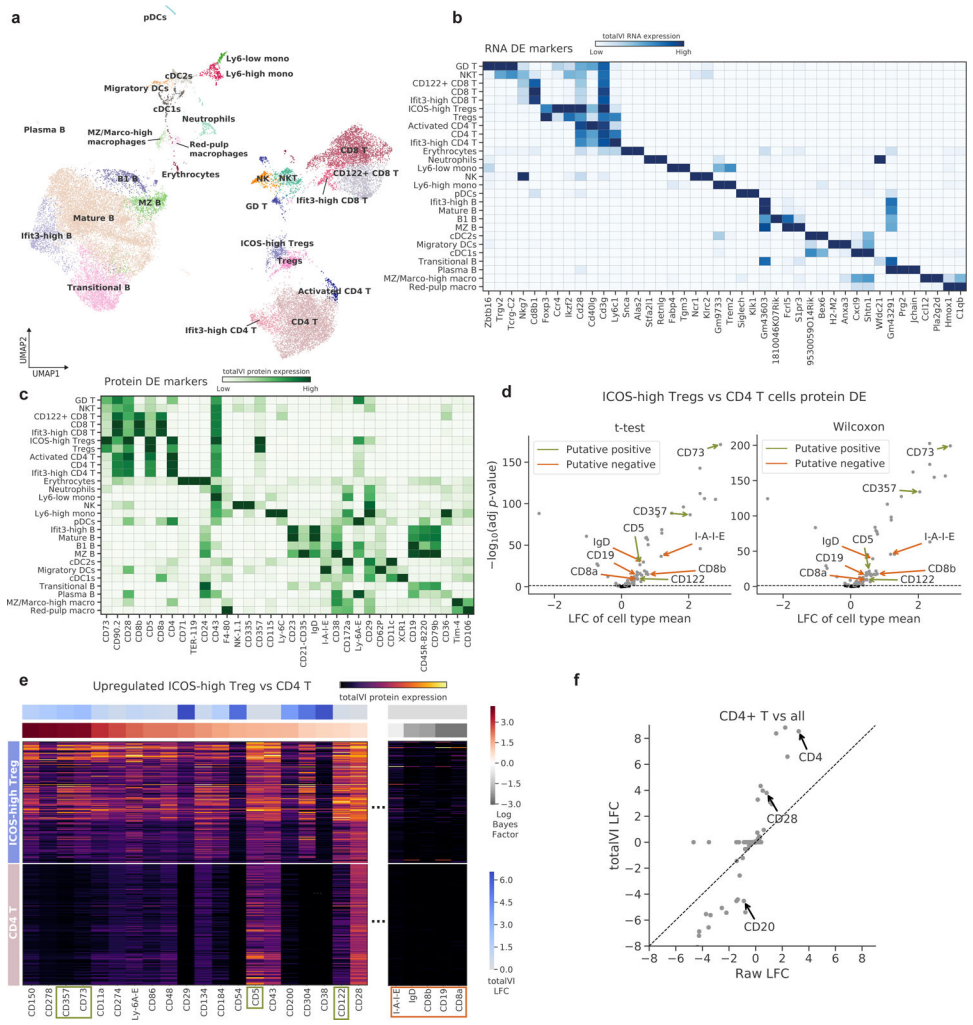


Figure 4: totalVI identifies differentially expressed genes and proteins.

totalVI intersect was applied to the SLN-all dataset. **a**, UMAP plot of SLN-all, after clustering and annotating the data (Methods 4.11). **b, c**, Heatmap of markers derived from one-vs-all tests for **(b)** RNA and **(c)** proteins. For each cell type, we display the top three protein markers and top two RNA markers in terms of LFC. **d**, Volcano plot of protein differential expression test between ICOS-high Tregs and CD4 T cells for a Welch's t-test and Wilcoxon rank-sum test. Putative positives and negatives are denoted by green and orange arrows, respectively. Significant proteins (BH-adjusted p-value < 0.05) are colored in grey, all others are in black. **e**, totalVI protein expression for proteins (columns) upregulated in ICOS-high Tregs versus CD4 T cells. Cells (rows) are ordered by cluster, and subsampled to be equal in number per cluster. Columns are normalized in the range [0, 1]. The left section in the heatmap contains all the proteins called differentially expressed by totalVI with a positive log fold change. Proteins are sorted by Bayes factor (significance). The rightmost section contains the putative negatives, which are not called differentially expressed by totalVI. **f**, Comparison of log fold changes estimated by totalVI and observed in the raw data from a one-vs-all test of CD4 T cells.

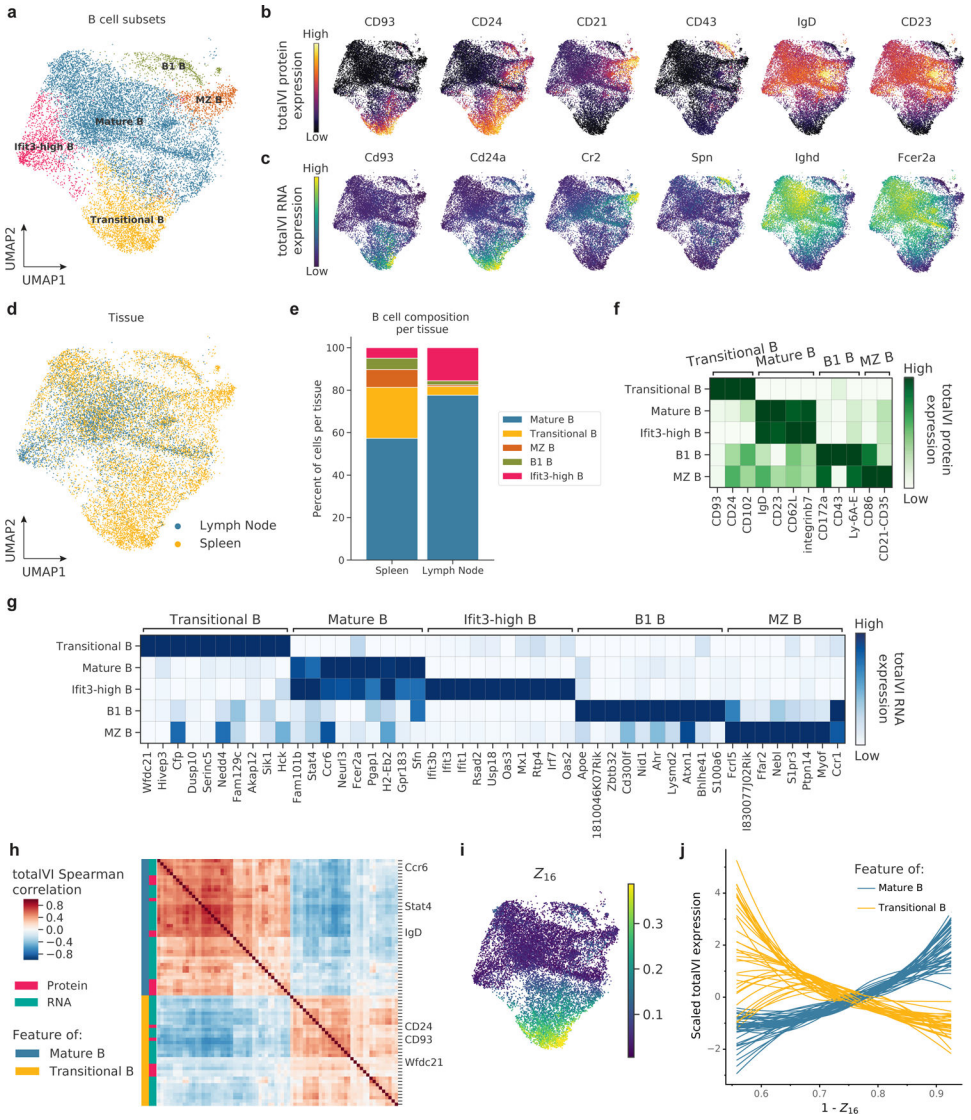


Figure 5: Characterization of B cell heterogeneity in the spleen and lymph nodes with RNA and protein.

totalVI-intersect was applied to the SLN-all dataset. Data were filtered to include B cells. **a**, UMAP plot of totalVI latent space labeled by cell type. **b, c**, UMAP plots of totalVI latent space colored by **(b)** totalVI protein expression of six marker proteins and **(c)** totalVI RNA expression of the six genes that encode the corresponding proteins in (b). **d**, UMAP plot of totalVI latent space labeled by tissue. **e**, Cell type composition per tissue. **f, g**, totalVI one-vs-all differential expression test on B cell subsets filtered for significance (Methods) and sorted by the totalVI median LFC. **(f)** The top three differentially expressed proteins per subset and **(g)** the top ten differentially expressed genes per subset, arranged by the subset in which the LFC is highest. **h**, totalVI Spearman correlations in transitional B cells between RNA and proteins, which were selected as described in Methods. Features were hierarchically clustered and are labeled as either RNA or protein, and by the cell type with which the feature is associated. **i**, UMAP plot of totalVI latent space colored by Z_{16} (the totalVI latent dimension associated with transitional B cells). **j**, totalVI expression of

features in (h) as a function of $(1 - Z_{16})$. Each feature was standard scaled and smoothed with a loess curve.