Integrating single-cell RNA-seq datasets with substantial batch effects

Karin Hrovatin[1,2,3,4], Amir Ali Moinfar[1,5], Alejandro Tejada Lapuerta[1,5], Luke Zappia[1,5], Ben Lengerich[3,4], Manolis Kellis[3,4], Fabian J. Theis[1,2,5]*

1: Institute of Computational Biology, Helmholtz Zentrum München, Neuherberg, Germany.
2: TUM School of Life Sciences Weihenstephan, Technical University of Munich, Freising, Germany.
3: Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA.
4: Broad Institute of MIT and Harvard, Cambridge, MA.
5: School of Computation, Information and Technology, Technical University of Munich, Garching, Germany.

*Correspondance should be addressed to F.J.T.: fabian.theis@helmholtz-muenchen.de

Abstract

Computational methods for integrating scRNA-seq datasets often struggle to harmonize datasets with substantial differences driven by technical or biological variation, such as between different species, organoids and primary tissue, or different scRNA-seq protocols, including single-cell and single-nuclei. Given that many widely adopted and scalable methods are based on conditional variational autoencoders (cVAE), we hypothesize that machine learning interventions to standard cVAEs can help improve batch effect removal while potentially preserving biological variation more effectively. To address this, we assess four strategies applied to commonly used cVAE models: the previously proposed Kullback–Leibler divergence (KL) regularization tuning and adversarial learning, as well as cycle-consistency loss (previously applied to multi-omic integration) and the multimodal variational mixture of posteriors prior (VampPrior) that has not yet been applied to integration. We evaluated performance in three data settings, namely cross-species, organoid-tissue, and cell-nuclei integration. Cycle-consistency and VampPrior improved batch correction while retaining high biological preservation, with their combination further increasing performance. While adversarial learning led to the strongest batch correction, its preservation of within-cell type variation did not match that of VampPrior or cycle-consistency models, and it was also prone to mixing unrelated cell types with different proportions across batches. KL regularization strength tuning had the least favorable performance, as it jointly removed biological and batch variation by reducing the number of effectively used embedding dimensions. Based on our findings, we recommend the adoption of the VampPrior in combination with the cycle-consistency loss for integrating datasets with substantial batch effects.

Keywords

single-cell RNA sequencing (scRNA-seq), data integration, VampPrior, latent cycle-consistency, KL regularization strength, adversarial learning, benchmarking

1. Introduction

Joint analysis of multiple single-cell RNA sequencing (scRNA-seq) datasets has recently provided new insights that could not have been obtained from individual datasets. For example, pooling of datasets generated in different studies enabled cross-condition comparisons[1,2], population-level analysis[3,4], and revealed evolutionary relationships between individual cell types[5]. The selection of pre-clinical models, such as organoids and animals, and the characterization of their limitations likewise rely on comparison with human tissues[6–12]. Similarly, data generated with different protocols was compared to evaluate protocol usability[13,14]. With the increasing number of publicly available scRNA-seq datasets[15] the number of such cross-dataset analyses is also increasing.

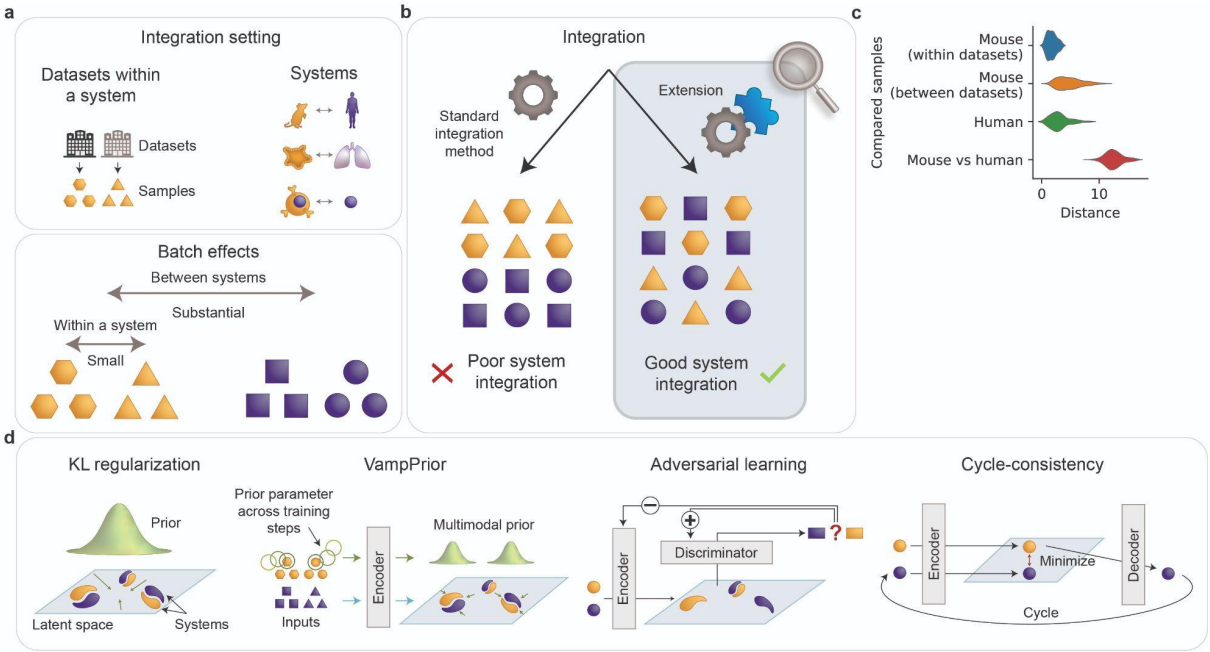


Figure 1: The challenge of integrating datasets with substantial batch effects. (**a**) Substantial batch effects are present between different biological "systems", such as cross-species, organoid-tissue, and cell-nuclei datasets. (**b**) Integrating datasets with substantial batch effects poses a bigger challenge than integrating datasets of similar samples across laboratories, where the batch effect is smaller. In this study, we evaluate different approaches for improving cVAE-based batch correction. (**c**) Pre-integration distances between samples within a dataset, samples between datasets within a system, and samples between different systems. Euclidean distance distributions in the PCA space of the mouse-human dataset between mean embeddings of delta cells are shown as an example. The significance of differences between within and between systems distance distributions is reported in **Supplementary Table S1** for all cell types and datasets. (**d**) Overview of approaches for increasing batch removal strength in cVAE-based models: KL-loss-based regularization of the latent space, the use of the VampPrior as a replacement for the standard Gaussian prior, and adversarial learning and cycle-consistency loss that actively push together samples from different systems. Parts of the cVAE model were omitted from individual panels for brevity. For a detailed description of a standard cVAE see the *Experimental setup* section.

These analyses can be complicated due to technical and biological differences[16–18]. To combat this, multiple computational methods for the integration of diverse datasets were developed[19,20]. Among the most popular and best-performing methods are conditional variational autoencoder (cVAE) based models, which can correct non-linear batch effects, are flexible in the choice of batch covariates, and are scalable to large datasets[17,19]. They work by adding batch information as a covariate to the encoder and decoder, thus enabling learning of a batch-free latent space[20,21]. These methods enable good integration of batch effects arising by processing similar samples in different laboratories, however, they do not enable sufficient integration when differences between datasets are more substantial due to datasets originating from distinct biological or technical "systems", such as multiple species or sequencing technologies (e.g. cell-nuclei)[16–18] (**Figure 1a,b**). To improve performance different approaches for

tuning and increasing batch correction have been proposed, such as Kullback–Leibler divergence (KL) regularization strength tuning[22], latent space adversarial learning[23–25], and latent space cycle-consistency that was previously used specifically for multi-omic integration[26–28], and for increasing biological preservation the use of the multimodal variational mixture of posteriors (VampPrior)[29] as the prior for the latent space was proposed[30]. However, it is unclear what the strengths and weaknesses of these approaches are and how they compare to each other when applied in different scenarios.

Here, we compare approaches for improving integration with cVAEs, including newly implemented cycle-consistency and VampPrior models and existing tools, such as scVI[20], which is based on standard cVAE, and GLUE[23], which uses adversarial learning (**Figure 1d**). We integrate datasets with substantial batch effects, including cross-species, organoid-tissue, and cell-nuclei datasets. We systematically evaluate batch correction and biological preservation on cell type and sub-cell type levels, with existing[17] and newly proposed metrics and additional downstream analyses. In order to better understand how integration performance can be customized, we perform extensive tuning of hyperparameters that regulate batch correction for all analyzed methods. In short, we find that cycle-consistency and VampPrior improve batch correction while retaining high biological preservation in contrast to a standard cVAE model where increased batch correction directly leads to lower biological preservation.

2. Experimental setup

2.1. Datasets with substantial batch effects caused by biological and technical factors

To evaluate cross-system integration we selected multiple datasets where batch effects are more substantial than what is commonly observed between samples within a dataset or between similar datasets generated in different laboratories. This includes three data settings: organoids and primary tissue (organoid-tissue; retinal organoids and adult human retina samples), scRNA-seq and single-nuclei RNA-seq (snRNA-seq) (cell-nuclei; subcutaneous adipose tissue profiled with both technologies), and different species (mouse-human; pancreatic islets from mouse and human). We confirmed that in all three settings, the per-cell type distances between batches on non-integrated data are significantly smaller within systems, both within and between datasets, than between systems (**Figure 1c**, **Supplementary Table S1**).

2.2. Approaches for improving batch effect correction with variational autoencoders

In cVAEs, the encoder (E_{ϕ}) embeds cell expression (x) and batch information (c), into a batch-effect corrected latent representation (z). Then, a decoder (G_{θ}) reconstructs the expression based on latent representation and batch information. Two opposing losses are used to train the model (combined as $L_{cVAE(\theta,\phi)}$), the expression reconstruction loss that promotes information preservation during encoding and decoding and KL loss-based regularization of latent space that encourages information compression towards a Gaussian prior distribution.

$$L_{cVAE(\theta,\phi)} = -\mathbb{E}_{z\sim q_\phi(z|x,c)}[\log p_\theta(x|z,c)] + D_{KL}(q_\phi(z|x,c)||p_\theta(z))$$

Below, we describe different approaches for improving integration by modifying the VAE objective (**Figure 1d**).

2.2.1. Tuning of KL regularization strength in cVAEs

The most straightforward approach for increasing integration strength is by using a higher KL regularization loss weight. This leads to lower preservation of information within the latent space, as it pushes samples' latent representations towards the Gaussian prior distribution. The information is removed not only for technical or batch variation, which is desired but as a side effect also for biological variation[22]. To enable good initialization of biological representation the KL regularization loss weight can be gradually increased during training via annealing, as done in scvi-tools[31]. Thus, we tested integrations with different KL regularization loss weights on our implementation of standard cVAE and its implementation in scvi-tools[20], which is among the most popular integration models.

2.2.2. Adversarial loss

An additional adversarial classification loss (L_{ADV}) can be added to the L_{cVAE} objective that promotes indistinguishability of latent embeddings from different batches, with the number of batches K[23–25]. For this, a

discriminator ($D_{k,\psi}$) is added, which is trained by minimizing classification loss (L_{ADV}), while the encoder of cVAE is trained by maximizing L_{ADV}, opposing the discriminator's ability to distinguish between batches. As an example of a model that leverages adversarial learning for batch correction, we used GLUE[23], where we tuned L_{ADV} loss weight and other losses that compete with L_{ADV}.

$$L_{ADV(\phi,\psi)} = \frac{1}{K}\sum_{k=1}^{K} E_{z\sim q_\phi(z|x,c)}[CrossEntropy(D_k(z;\psi), c_k)]$$

2.2.3. Cycle-consistency loss

Latent space cycle-consistency works by pushing together the latent representation of matched cells from two batches via an additional loss component (L_{CYC}). Cell x_i belonging to the batch i is encoded into a latent representation z_i. That latent representation is then decoded with batch covariate j into the cell x'_j, which represents the cell x_i as if it originated from the batch j. The cell x'_j is encoded into the latent representation z'_j. Finally, the model penalizes the distance between the original representation z_i and the cycle-generated z'_j. Here, different distance minimization losses can be applied, with our choice being the mean squared error (MSE) on standardized data.

$$z_i = E_\phi(x_i, c_i)$$

$$z'_j = E_\phi(G_\theta(z_i, c_j), c_j)$$

$$L_{CYC(\theta,\psi)} = E_{z\sim q_\phi(z|x,c)}[Distance(z_i, z'_j)]$$

This approach has previously been used for integration across modalities[26,28], but it has not been tested for integration of substantial batch effects within RNA-seq data. We added L_{CYC} to our implementation of cVAE to assess how the integration performs when tuning the L_{CYC} loss weight.

2.2.4. The VampPrior

The VampPrior was initially proposed for generating more expressive latent representations due to the use of multiple prior components[29]. It was previously shown that this also applies to single-cell data[30], similar to other flexible priors, such as standard Gaussian mixtures (GM)[32–34], and may be used to add prior information to latent space regularization[35]. However, the VampPrior has not previously been used for improving batch integration.

The VampPrior replaces the unimodal Gaussian prior $p_\theta(z)$ with a mixture of trainable Gaussian prior components (with the number of components L). Their parameters are not defined in the latent space, as is usually done in cVAEs, but rather in the input cell space as "pseudoinputs" (xpi_l), which are passed through the encoder to obtain latent representations (mean, variance) used to parametrize the Gaussian components of the prior distribution. This thus directly couples the prior and the posterior. Additionally, the weights (w) of the prior components are likewise learned[29,36].

$$p_\theta(z) = \sum_{l=1}^{L} Softmax(w_l) q_\phi(z|xpi_l)$$

We tested VampPrior in our cVAE implementation and inspected how the number of prior components affects performance, also in comparison to using a simpler GM prior. The model with the VampPrior can be likewise as standard cVAE combined with other approaches for increasing batch correction. Thus, we tested both tuning of the KL regularization strength as well as combined the model with cycle-consistency loss.

2.3. Integration evaluation

We compared the biological preservation and batch correction performance of existing tools and newly proposed models for cross-system integration. We tuned individual model hyperparameters that are used for directly regulating batch correction, as described above, and ran every model setting multiple times with different random seeds to capture random variation in model performance. To evaluate integration quality, we measured system-level batch correction with graph integration local inverse Simpson's index (iLISI), which assesses batch mixing in the local neighborhood of every cell, coarse biological preservation via preservation of cell type clusters with normalized mutual information (NMI), which compares ground-truth cell type labels to clusters in the integrated embedding, and fine biological preservation via the preservation of within-cell type variation with Moran's I, which measures preservation of gene expression patterns within samples in the integrated embedding. For all metrics higher scores indicate better performance. We further describe the rationale for metric selection in **Supplementary Note S1**.

3. Results

3.1. Existing models struggle with under-integration and loss of biological information

Standard cVAE, used as a baseline, had poor batch correction across all tested datasets, in comparison to other models, and the systems were clearly visually separated on the UMAP (**Figure 2**, **Supplementary Figure S1**, **Supplementary Figure S2**), indicating that it alone is unsuitable for integrating substantial batch effects. We also observed that increased KL regularization strength led to stronger batch correction due to fewer embedding dimensions that were effectively used in downstream analyses due to dimension shrinkage (i.e. some latent dimensions being set close to zero in all cells) resulting in information loss (**Supplementary Figure S4**). This was reflected in the removal of the main KL regularization-induced batch correction and biological preservation changes after standardizing individual embedding features (**Figure 3**, **Supplementary Note S3**). Thus, KL weight is a less favorable approach for removing batch effects as it jointly, without discrimination, removes both biological and batch variation. In comparison, the cVAE-based model scVI[20], which is regarded as a state-of-the-art model for scRNA-seq integration[17], consistently had the best biological preservation. However, it had a relatively low batch correction compared to other models, especially in the mouse-human and organoid-tissue datasets that were harder to integrate than the cell-nuclei dataset due to more substantial batch effects (**Supplementary Figure S5**).

Next, we analyzed the performance of an existing model that uses adversarial learning (GLUE[23]). GLUE had the highest batch correction overall and in most cases, comparable cell type preservation (NMI) to other models, however, it had lower performance on finer within cell type variation (Moran's I). Thus, adversarial learning may be less suited when the aim of the integration includes downstream cell state analyses, as indicated by the partial loss of gene group variability patterns known to be present in the analyzed data (**Supplementary Figure S7c**). Moreover, adversarial learning also led to the mixing of unrelated cell types with differences in proportions across systems (e.g. delta, acinar, and immune cells in the mouse-human pancreatic data, **Figure 2b**), even though they were easily separated by other models. In **Supplementary Note S4** we further explain how disregard of cell identities in adversarial learning leads to the above-mentioned issues. Therefore, adversarial learning is problematic as a general integration strategy as datasets often have unbalanced cell proportions (**Supplementary Figure S8**) due to biological differences or differences in technical protocol capture[37,38].

Furthermore, one important characteristic of integration models is also to not over-integrate (i.e., align cell populations across systems that are biologically distinct). The model can avoid over-integration in two ways. First, if dissimilar cell types are embedded in the same region of the latent space this will lead to higher reconstruction loss, thus balancing out the over-integration. Second, the relevant biological variation and undesired batch effects between systems may be disentangled if the batch effects are consistent across cell types, while system-specific biological variation is often cell-type-specific. Models like scGEN[39], which force embedding overlap on per-cell type level and do not consider reconstruction capability hitherto, lead to over-integration if the input batch covariate is not orthogonal to biological variation, as is usually the case (**Supplementary Note S5**). Therefore, as scGEN batch correction depends completely on the input batch covariate rather than being learned, we did not include scGEN in our benchmark. In terms of sample alignment performance, all models correctly separated retinal pigment epithelial cells between organoid and tissue data[7] (**Supplementary Figure S2a**). However, finer cross-system

differences were lost in some models, such as GLUE, leading to co-localization of Mueller cells from the tissue fovea and periphery with organoid cells (**Figure 2c, Supplementary Figure S3**), despite the *RHCG* expressing fovea cells having been reported to be distinct from the periphery and organoid cells[7].

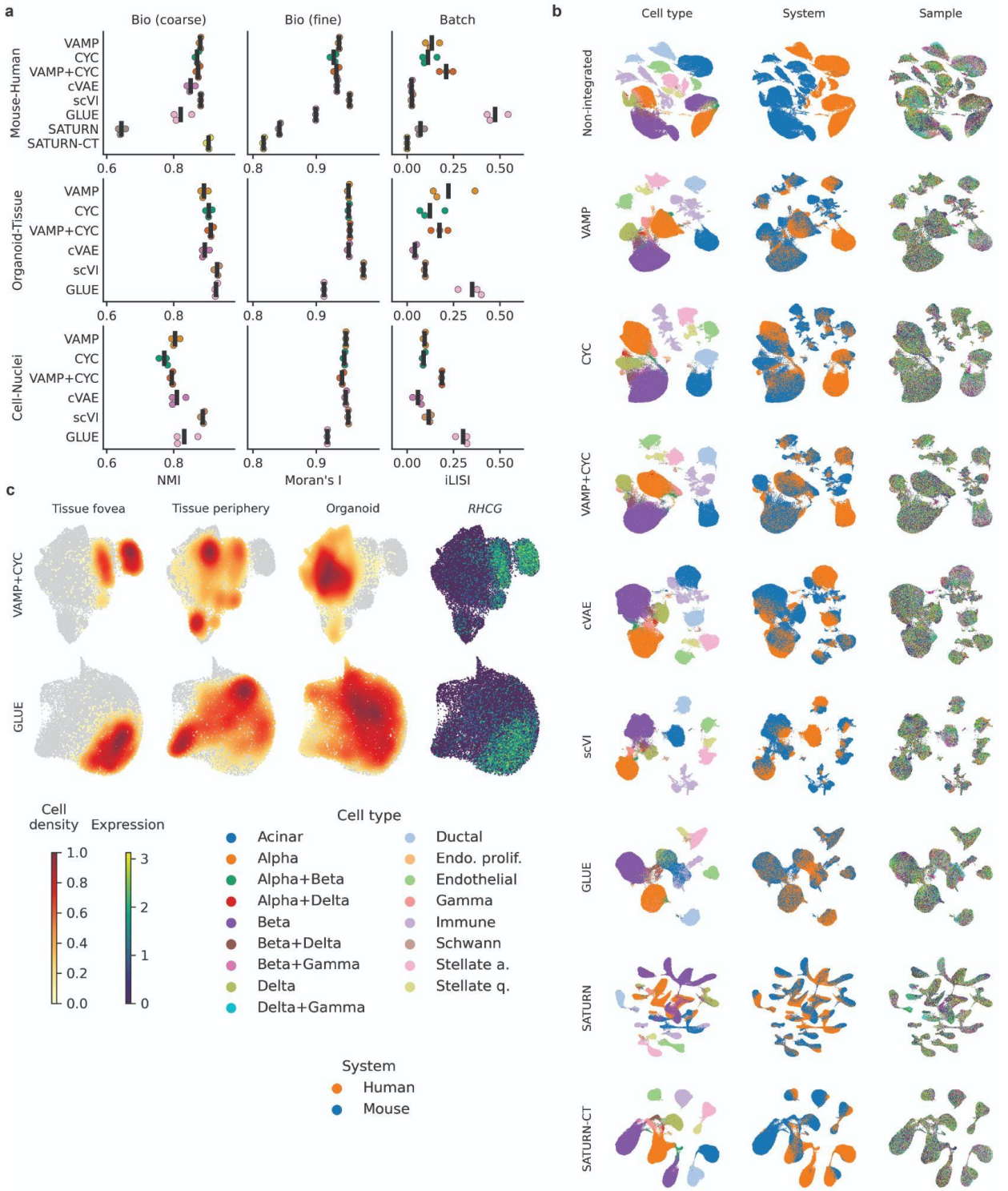


Figure 2: While existing models struggle with over- or under-integration, the VAMP+CYC model achieves a good trade-off between batch correction and biological preservation. (**a**) We selected the best hyperparameter settings

based on combined biological preservation and batch correction performance for every model, with different integration metrics in columns (larger values indicate better integration performance) and cross-system data settings in rows, showing the average performance (vertical line) of three runs (dots). Results of all hyperparameter settings are in **Supplementary Figure S1**. (**b**) UMAPs of representative runs for the best hyperparameter setting in the mouse-human dataset. A legend for sample colors is not shown due to the large number of samples. (**c**) UMAPs of Mueller cells based on representative runs for the best hyperparameter setting in the organoid-tissue dataset. The UMAPs are colored by the cell density of organoid and tissue periphery cells, which are expected to overlap, and of tissue fovea cells expressing *RHCG*, which are expected to separate from the other two cell populations. Results for all models are shown in **Supplementary Figure S3**.

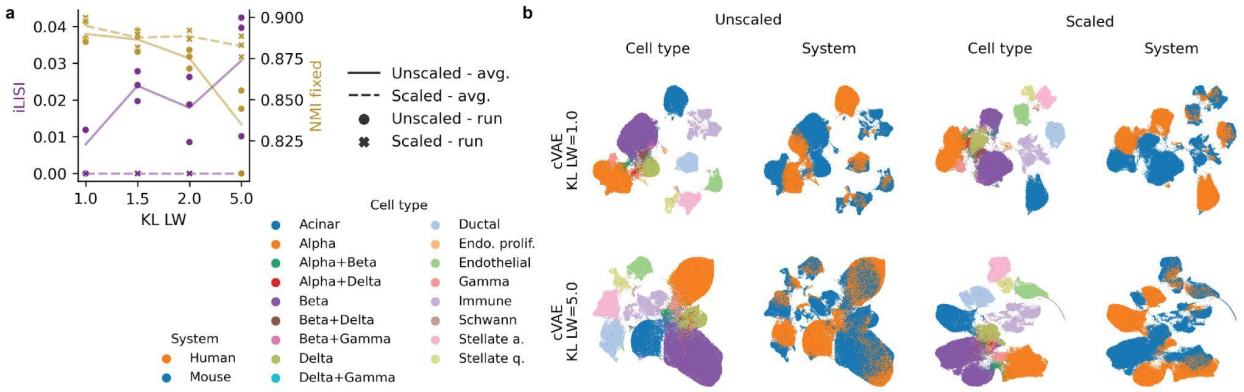


Figure 3: Scaling of cVAE-based embedding results in higher biological and batch information preservation across KL regularization strengths. (**a**) Integration metrics (y-axes) on scaled and unscaled cVAE embedding with different KL regularization loss weights (KL LW). Shown are individual runs with different seeds (points) and their averages (avg., lines). We used a different version of NMI metric (NMI fixed) to better distinguish between over- and under-integration (see **Supplementary Note S1**). (**b**) UMAP visualization of scaled and unscaled cVAE mouse-human embedding obtained with low or high KL regularization strength (shown runs with seed=1). As seen in (**a**), scaling especially affects biological preservation and batch correction at stronger KL regularization, reflected in UMAPs as crisper cell type clusters and reduced system mixing. UMAPs for other datasets are shown in **Supplementary Figure S6**.

As one of our settings was cross-species integration, which is complicated by different gene sets across species, we further assessed models that accept different input dimensions across systems, namely GLUE and SATURN[40], which is an example of a model designed specifically for cross-species integration. We ran both models with one-to-one orthologues (OTO) as well as with a flexible orthology (FO) gene set that included different types of orthologues and non-orthologous genes. Integration with FO input genes did not outperform the OTO approach in either GLUE or SATURN (**Supplementary Figure S1**), indicating that more advanced gene mapping may not be needed for mouse-human integration. However, for more divergent species or species with lower-quality annotation and orthologue mapping, using FO genes could still be beneficial[16]. Furthermore, as SATURN requires prior cell type labels for every system, we ran it with ground-truth cell type labels (SATURN-CT) as well as clustering-based labels (SATURN). However, ground-truth labels are often not available before integration and could also lead to a positive bias in the NMI metric. Both SATURN models performed poorly in biological preservation and batch correction, with the expected positive NMI bias in SATURN-CT, while Moran's I remained low. The poor biological preservation can be explained by over-reliance on prior cell cluster labels, as discussed in **Supplementary Note S6**.

3.2. Multimodal priors with cycle-consistency improve integration

To address the issues of the above-mentioned models we extended the cVAE model with cycle-consistency (CYC), the VampPrior (VAMP), and cycle-consistency and the VampPrior (VAMP+CYC). These models consistently outperformed the base cVAE in batch correction while retaining similar BioC. The VAMP+CYC was also able to achieve higher batch correction at comparable BioC than CYC and VAMP alone (**Supplementary Figure S1**). Thus, adding the VampPrior and cycle-consistency to other cVAE-based models, such as scVI, may further increase their performance. In comparison to GLUE, VAMP+CYC showed lower batch correction but had higher fine biological

preservation and was not prone to mixing of cell types with different proportions across systems (**Supplementary Note S4**), making it better balanced in terms of biological preservation and batch correction. Furthermore, it correctly aligned Mueller cells from tissue periphery and organoids while separating cells from tissue fovea (**Figure 2c**), indicating its ability to account for biological differences across systems during integration.

As the VampPrior has not been used before for integration, we further investigated what is the cause of improved batch correction. We inspected the relationship between the prior and the data, which revealed that individual prior components covered different cell types rather than systems (**Supplementary Figure S9**). To test if batch correction is improved due to the multimodal prior, we added a VampPrior model with only one prior component that resembles a normal cVAE, but in contrast to cVAE the prior parameters are learned, and a GM prior model (GMM) as a simpler multimodal prior alternative where prior and posterior distributions are not coupled. We also tested if trainable prior components are key for integration by fixing prior component parameters. The batch correction was much lower when using only one prior component, compared to two or more, and similarly, batch correction was increased in GMM with multiple prior components. Fixing of prior components had little effect on VAMP performance (**Figure 4**, **Supplementary Figure S10**) and likewise the initialization stratified by cell type or system did not affect performance (**Supplementary Figure S11**). This indicates that the number of prior components is key for improved batch correction. Nevertheless, VAMP outperformed GMM both in the achieved batch correction as well as in the robustness to a different number of prior components. This indicates that not only the multimodal prior, but also the coupling of the prior and the posterior improves integration performance. We further describe how the initialization and learning of prior components affect VAMP and GMM in **Supplementary Note S7**.

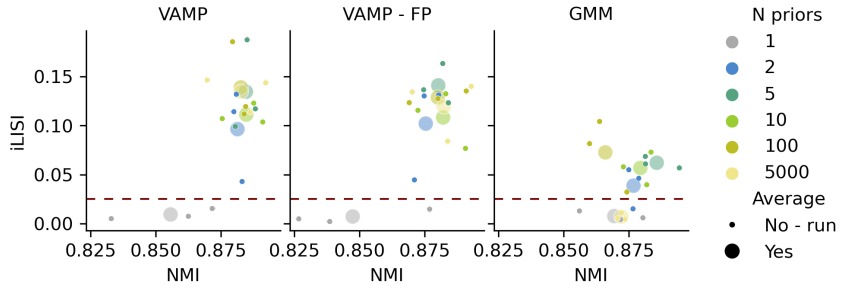


Figure 4: Effect of multimodal and learnable priors on integration. Scatter plots of NMI vs iLISI scores for different models (panels), including a VampPrior model with fixed prior components (FP). Points are colored by the number of prior components. Small points represent individual runs with different seeds and large points their average. The red line indicates separation in iLISI score between VAMP runs with a single and multiple prior components. Results are shown for the mouse-human dataset, for other datasets see **Supplementary Figure S10**.

3.3. Similar hyperparameter ranges may be optimal across datasets

Tuning of hyperparameters for batch correction strength regulation revealed similarities in optimal ranges across datasets (**Supplementary Figure S1**). Interestingly, we observed that for some batch correction regulation hyperparameters, the change in their value is not monotonously related to batch correction strength, as discussed in **Supplementary Note S2**.

4. Discussion

In order to improve the integration of scRNA-seq datasets with substantial batch effects we evaluated multiple cVAE-based strategies for increasing batch correction in three scenarios (cross-species, organoid-tissue, and cell-nuclei). We observed consistent model performance characteristics and failure modes across all three scenarios, enabling us to make recommendations regarding individual models.

The most straightforward and well-known approach, KL regularization strength tuning, jointly removed both biological variation and batch effects due to the reduced number of embedding dimensions effectively used in downstream analyses. Thus, this approach is highly inefficient for improving batch correction. We evaluated two alternatives, adversarial learning and cycle-consistency. While adversarial learning led to the highest batch

correction, it was also prone to mixing unrelated cell types with different proportions across systems due to not considering cell identities. Adversarial learning also led to lower within-cell type biological preservation. Therefore, adversarial learning should be used with caution as cross-system scRNA-seq datasets usually contain different cell type proportions and downstream applications often involve within-cell type analyses, for example, to inspect aging or disease-related effects. In contrast, cycle-consistency loss tuning led to increased batch correction, but was not prone to mixing cells from unrelated cell types and subtypes, making it a more promising option than adversarial learning.

We also examined the VampPrior and found that it led not only to improved biological preservation but also increased batch correction. A similar effect was also observed when using a simpler GM prior. However, the VampPrior showed overall better performance and higher robustness to varying numbers of prior components. Further work will be needed to fully understand the mechanisms of this phenomenon. These results may not only have applications in scRNA-seq integration but also in other domains using cVAE models for covariate effect removal. The performance of the model with the VampPrior was further improved when adding cycle-consistency, making this combination our method of choice for integrating datasets with substantial batch effects. However, if batch effects are not substantial commonly used methods like scVI may be sufficient for integration.

The here-described extensions for improving integration could be in the future easily added to other cVAE-based integration tools, such as scVI. This could also extend the usability and performance of our VAMP+CYC implementation, for example by enabling sample-level batch effect representation learning in combination with scPoli[41], which eases integration of population-wide datasets, or label-aware integration in combination with scANVI[42], which due to supervision outperforms other models such as scVI[17].

Lastly, our results reveal optimal hyperparameter ranges across datasets, easing the setting of default values or hyperparameter tuning. However, the exact optima are nevertheless dataset-dependent. We show that tuning of hyperparameters sometimes produced unexpected results, for example, batch correction was in some cases not monotonously related to the hyperparameter responsible for batch correction strength regulation.

In conclusion, due to the excellent performance of the combined VampPrior and cycle-consistency model, we advocate this strategy as a general improvement to common integration methods. We showed that this strategy produces high-quality embeddings which lead to biologically truthful conclusions in downstream analyses.

5. Extended methods

5.1. Data preprocessing

Data for each of the three data settings was prepared separately as described below. For the mouse-human data setting we used pancreatic islet datasets of mouse[1] (without embryonic and low-quality cells) and human[43], for the organoid-tissue scenario we used a retinal dataset[7], and for the cell-nuclei scenario we used adipose dataset[44] (using the SAT fat type). We obtained published count data for all datasets (see Data availability section) and removed unannotated cells. In each system, we kept genes expressed in more than 20 cells, and in the mouse-human OTO setting we also only kept OTO genes, while in the FO setting, we removed genes without unique gene symbols (required for SATURN). Data was normalized with Scanpy *normalize_total* and log-transformed, and 3000 HVGs (4000 HVGs for organoid-tissue) were selected per system, keeping the union of HVGs across systems. For GLUE we computed a non-integrated principal component analysis (PCA) per system on scaled data using 15 principal components (PCs) and for SATURN we used this data to compute prior clusters per system.

Non-integrated embeddings were computed on the same cells as used for integration evaluation (described below). The normalized expression prepared for integration was standardized per gene, followed by computing 15 PCs, neighbors, and UMAP.

5.2. Evaluation of batch effect strength in unintegrated data

Batch effect strength comparison within and between systems within a data setting was performed by computing Euclidean distances between mean embeddings of cell type and sample groups in the PC space (15 PCs on scaled OTO data). We used only groups with at least 50 cells and cell types where both systems had at least three

remaining samples. The significance of differences in distance distributions within and between systems was computed per cell type with the one-sided Mann–Whitney U test.

Batch effect strength analysis of data settings was performed using average silhouette width (ASW) with systems as the batch covariates for individual cell types within every data setting, using the non-integrated embeddings. We adapted scIB metrics function[17,45] so that ASW scores were not reported as absolute values. We used ASW rather than iLISI metric as iLISI was not discriminative enough the substantial pre-integration batch effects. As the computation was performed on the PCA space with comparable dimension ranges across data settings, the dimension range bias described in **Supplementary Note S1** does not affect the comparison.

5.3. Integration

We corrected sample and system-level batch effects during integration by adding them to the model inputs as one-hot encoded vectors. The L_{CYC} and L_{ADV} were computed only on the system covariate. All existing models were run with default parameters, except for the following changes.

For scVI we used *n_layers*=2, *n_hidden*=256, *n_latent*=15, *n_steps_kl_warmup*=1600, and *gene_likelihood*=nb and we tuned *max_kl_weight* (KL loss weight). In GLUE we tuned *rel_gene_weight* (gene graph weights), *lam_graph* (graph loss weight), and *lam_align* (alignment loss weight). In SATURN we used the provided ESM2 protein embeddings and tuned *pe_sim_penalty* (protein similarity loss weight). The number of epochs was set to a fixed value per method and dataset, depending on the number of cells.

Our custom cVAE implementation was based on the scVI framework[31]. Unlike in scVI, we used the Gaussian log-likelihood on normalized log-transformed data for reconstruction loss and did not use KL weight annealing. We used the same number of layers and dimensions as for scVI. We added L_{CYC} and L_{ADV} on the standard cVAE model.

We implemented the VampPrior as described by the authors[29,46] and did likewise for the GM prior, but with prior components representing points in the latent space. For the VampPrior we initialized prior components by randomly sampling cells from the data and for GM prior we either used sampled data that we passed through the encoder before training or used random initialization with mean in the range [0,1) and variance of one. L_{CYC} was computed between the latent representation of a cell ($z_{c=i}$) and its cycle-pair coming from the other system ($z'_{c=j}$) using MSE on data standardized within a minibach separately for cells and their cycle pairs.

5.4. Integration evaluation

We performed evaluations on at most 100,000 randomly selected cells per dataset to reduce the computational cost, except for Moran's I where cells were selected as described below. We computed neighbors on the latent embedding directly except where we specified that the embedding dimensions were standardized prior to neighbors computation. This data was also used for UMAPs. The non-integrated embedding was computed with 15 PCs on scaled OTO data, using the same set of cells as for the integrated data.

For LISI and ASW metrics we used implementations from scib-metrics Python package[45] and for NMI we adapted their implementation to set a random seed. We computed the NMI-fixed and Jaccard index by first computing Leiden clusters at high resolution (r=2) and then assigning a cell type label to each cluster based on the most common ground-truth label. This annotation was used for comparison to ground-truth labels. For Moran's I we compared Moran's I values between non-integrated and integrated data as follows. For the pre-integration Moran's I computation we kept sample and cell type groups with at least 500 cells and excluded doublets. For each group, we removed genes expressed in less than 10% of the group's cells, computed 15 PCs and neighbors per group, and used them to compute Moran's I on all genes. We set a cutoff on Moran's I per dataset to keep from around a dozen to around 150 genes per group for integration evaluation. On integrated data, we then repeated the Moran's I calculation on the same cell groups and the selected genes. The final score (mi) was defined as a ratio of post- (mi_{post}) and pre-integration (mi_{pre}) values averaged across genes (G), samples (S), and cell types (CT).

$$mi = \mathrm{avg}_{CT}(\mathrm{avg}_{S}(\mathrm{avg}_{G}(\frac{mi_{post(G,S,CT)}}{mi_{pre(G,S,CT)}})))$$

We selected the best hyperparameter setting per model over all tested hyperparameters. For every model, we scaled individual metrics to [0,1] across all runs computed for hyperparameter optimization and then computed the biological preservation score as the average of Moran's I and NMI and batch correction as iLISI alone. The overall score was computed as a weighted average of biological preservation (weight=0.6) and batch correction (weight=0.4). The best hyperparameter setting was selected based on the average overall score across runs and for every hyperparameter setting we selected a representative run for UMAP plots as the run with the median overall score.

Preservation of gene groups known to be variable within pancreatic islet beta cells of healthy adult mice was evaluated with Moran's I on the control sample from the mSTZ dataset[1]. We used Scanpy *score_genes* to obtain a single score for every gene group and computed Moran's I on these scores for every embedding.

Data availability

The datasets were retrieved from public repositories: GEO (GSE211799), www.isletgenomics.org, Single Cell Portal (SCP1376), and https://cellxgene.cziscience.com/collections/2f4c738f-e2f3-4553-9db2-0582a38ea4dc.

Code availability

The code and the conda environments are available at: https://github.com/theislab/cross_system_integration.

Acknowledgments

We thank members of Theis lab for providing feedback, especially Sabrina Richter. This work was supported by funds from the Helmholtz Association and Helmholtz Munich. F.J.T. acknowledges support from the European Union (ERC, DeepCell—101054957, BetaRegeneration—101054564). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. K.H. acknowledges financial support from Joachim Herz Stiftung via Add-on Fellowships for Interdisciplinary Life Science and support from Helmholtz Association under the joint research school 'Munich School for Data Science'. M.K. acknowledges support from grants AG081017, NS129032, NS115064, AG074003, NS127187, and AG067151. B.L. acknowledges support from Alana Down Syndrome Center. A.T.L. acknowledges support from ERC Grant DeepCell.

Author contributions

Conception, data curation, and illustrations: K.H.; methodology, formal analysis, figures, and original paper draft: K.H. and A.A.M.; supervision: A.T.L., L.Z., B.L., M.K., F.J.T.; funding acquisition: M.K., F.J.T; paper review: all authors.

Competing interests

F.J.T. consults for Immunai Inc., Singularity Bio B.V., CytoReason Ltd and Cellarity; and has an ownership interest in Dermagnostix GmbH and Cellarity. L.Z. has consulted for Lamin Labs GmbH. The remaining authors declare no competing interests.

References

1. Hrovatin, K. *et al.* Delineating mouse β-cell identity during lifetime and in diabetes with a single cell atlas. *bioRxiv* 2022.12.22.521557 (2022) doi:10.1101/2022.12.22.521557.
2. Buechler, M. B. *et al.* Cross-tissue organization of the fibroblast lineage. *Nature* **593**, 575–579 (2021).
3. Nieto, P. *et al.* A single-cell tumor immune atlas for precision oncology. *Genome Res.* **31**, 1913–1926 (2021).
4. Sikkema, L. *et al.* An integrated cell atlas of the lung in health and disease. *Nat. Med.* **29**, 1563–1577 (2023).
5. Tarashansky, A. J. *et al.* Mapping single-cell atlases throughout Metazoa unravels cell type evolution. *Elife* **10**, (2021).
6. Tritschler, S. *et al.* A transcriptional cross species map of pancreatic islet cells. *Mol Metab* **66**, 101595 (2022).
7. Cowan, C. S. *et al.* Cell Types of the Human Retina and Its Organoids at Single-Cell Resolution. *Cell* **182**, 1623–1640.e34 (2020).
8. Sridhar, A. *et al.* Single-Cell Transcriptomic Comparison of Human Fetal Retina, hPSC-Derived Retinal Organoids, and Long-Term Retinal Cultures. *Cell Rep.* **30**, 1644–1659.e4 (2020).
9. Georges, A. *et al.* Comparing the transcriptome of developing native and iPSC-derived mouse retinae by single cell RNA sequencing. *Sci. Rep.* **13**, 1223 (2023).
10. Balboa, D. *et al.* Functional, metabolic and transcriptional maturation of human pancreatic islets derived from stem cells. *Nat. Biotechnol.* **40**, 1042–1055 (2022).
11. Wu, S. *et al.* Single-cell transcriptome in silico analysis reveals conserved regulatory programs in macrophages/monocytes of abdominal aortic aneurysm from multiple mouse models and human. *Front Cardiovasc Med* **9**, 1062106 (2022).
12. Ton, M.-L. N. *et al.* An atlas of rabbit development as a model for single-cell comparative genomics. *Nat. Cell Biol.* **25**, 1061–1072 (2023).
13. Basile, G. *et al.* Using single-nucleus RNA-sequencing to interrogate transcriptomic profiles of archived human pancreatic islets. *Genome Med.* **13**, 128 (2021).
14. Lake, B. B. *et al.* A comparative strategy for single-nucleus and single-cell transcriptomes confirms accuracy in predicted cell-type expression from nuclear RNA. *Sci. Rep.* **7**, 6031 (2017).
15. Svensson, V., da Veiga Beltrame, E. & Pachter, L. A curated database reveals trends in single-cell transcriptomics. *Database* **2020**, (2020).
16. Song, Y., Miao, Z., Brazma, A. & Papatheodorou, I. Benchmarking strategies for cross-species integration of single-cell RNA sequencing data. *bioRxiv* 2022.09.27.509674 (2023) doi:10.1101/2022.09.27.509674.
17. Luecken, M. D. *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).
18. Lütge, A. *et al.* CellMixS: quantifying and visualizing batch effects in single-cell RNA-seq data. *Life Sci Alliance* **4**, (2021).
19. Ryu, Y., Han, G. H., Jung, E. & Hwang, D. Integration of Single-Cell RNA-Seq Datasets: A Review of Computational Methods. *Mol. Cells* **46**, 106–119 (2023).
20. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
21. Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Alexander Wolf, F. Conditional out-of-distribution generation for unpaired data using transfer VAE. *Bioinformatics* vol. 36 i610–i617 Preprint at https://doi.org/10.1093/bioinformatics/btaa800 (2020).
22. Lotfollahi, M. *et al.* Biologically informed deep learning to query gene programs in single-cell

atlases. *Nat. Cell Biol.* **25**, 337–350 (2023).
23. Cao, Z.-J. & Gao, G. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nat. Biotechnol.* **40**, 1458–1466 (2022).
24. Lotfollahi, M. *et al.* Predicting cellular responses to complex perturbations in high-throughput screens. *Mol. Syst. Biol.* **19**, e11517 (2023).
25. Kopp, W., Akalin, A. & Ohler, U. Simultaneous dimensionality reduction and integration for single-cell ATAC-seq data using deep learning. *Nature Machine Intelligence* **4**, 162–168 (2022).
26. Wang, X. *et al.* Con-AAE: contrastive cycle adversarial autoencoders for single-cell multi-omics alignment and integration. *Bioinformatics* **39**, (2023).
27. Brombacher, E., Hackenberg, M., Kreutz, C., Binder, H. & Treppner, M. *The Performance of Deep Generative Models for Learning Joint Embeddings of Single-cell Multi-omics Data*. (Universität, 2022).
28. Xu, Y., Begoli, E. & McCord, R. P. sciCAN: single-cell chromatin accessibility and gene expression data integration via cycle-consistent adversarial network. *NPJ Syst Biol Appl* **8**, 33 (2022).
29. Tomczak, J. M. & Welling, M. VAE with a VampPrior. *arXiv [cs.LG]* (2017).
30. Dony, L., König, M., Fischer, D. S. & Theis, F. J. Variational autoencoders with flexible priors enable robust distribution learning on single-cell RNA sequencing data. https://icml-compbio.github.io/icml-website-2020/2020/papers/WCBICML2020_paper_37.pdf.
31. Gayoso, A. *et al.* A Python library for probabilistic analysis of single-cell omics data. *Nat. Biotechnol.* **40**, 163–166 (2022).
32. Grønbech, C. H. *et al.* scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics* **36**, 4415–4422 (2020).
33. Xu, J. *et al.* Graph embedding and Gaussian mixture variational autoencoder network for end-to-end analysis of single-cell RNA sequencing data. *Cell Rep Methods* **3**, 100382 (2023).
34. Kopf, A., Fortuin, V., Somnath, V. R. & Claassen, M. Mixture-of-Experts Variational Autoencoder for clustering and generating from similarity-based representations on single cell data. *PLoS Comput. Biol.* **17**, e1009086 (2021).
35. Lopez, R. *et al.* DestVI identifies continuums of cell types in spatial transcriptomics data. *Nat. Biotechnol.* **40**, 1360–1369 (2022).
36. Alemi, A. *et al.* An information-theoretic analysis of deep latent-variable models. (2018).
37. Denisenko, E. *et al.* Systematic assessment of tissue dissociation and storage biases in single-cell and single-nucleus RNA-seq workflows. *Genome Biol.* **21**, 130 (2020).
38. Zhao, Z. *et al.* Organoids. *Nat Rev Methods Primers* **2**, (2022).
39. Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nat. Methods* **16**, 715–721 (2019).
40. Rosen, Y. *et al.* Towards Universal Cell Embeddings: Integrating Single-cell RNA-seq Datasets across Species with SATURN. *bioRxiv* (2023) doi:10.1101/2023.02.03.526939.
41. De Donno, C. *et al.* Population-level integration of single-cell datasets enables multi-scale analysis across samples. *bioRxiv* 2022.11.28.517803 (2022) doi:10.1101/2022.11.28.517803.
42. Xu, C. *et al.* Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Mol. Syst. Biol.* **17**, e9620 (2021).
43. Elgamal, R. M. *et al.* An integrated map of cell type-specific gene expression in pancreatic islets. *bioRxiv* (2023) doi:10.1101/2023.02.03.526994.
44. Emont, M. P. *et al.* A single-cell atlas of human and mouse white adipose tissue. *Nature* **603**, 926–933 (2022).
45. *scib-metrics: Accelerated, Python-only, single-cell integration benchmarking metrics*. (Github).
46. 7_VAE_priors. https://jmtomczak.github.io/blog/7/7_priors.html.

Supplementary material

1. Supplementary notes

1.1. Supplementary Note S1: Selection of integration metrics

Recently, multiple metrics for the evaluation of integration methods were proposed and implemented in the scIB package[1]. However, we decided to use only a subset that we believe enables a higher-quality evaluation. We observed that metrics that operate on distances directly (such as ASW) rather than on the graph (such as LISI and NMI/ARI) lead to biases when more or less latent dimensions have a high-value range. We observed that when tuning integration hyperparameters, which leads to shrinkage of some latent dimensions (**Supplementary Figure S4**), metrics such as ASW led to inaccurate results that are in the most extreme cases even opposite to graph-based metrics and visually distinct patterns on UMAPs. To validate this, we simulated data (**Supplementary Figure S7a**) where one dimension had a bimodal distribution to represent two distinct groups and 15 additional dimensions represented Gaussian noise unrelated to the two groups. As a starting point, all noise dimensions had the same standard deviation. We then progressively shrank some of the noise dimensions to have a smaller standard deviation, thus corresponding to dimension shrinkage observed when tuning integration hyperparameters. When increasing the number of shrunken noise dimensions, but keeping the group dimension constant, the ASW-based metrics were much more strongly affected than the LISI-based metrics (**Supplementary Figure S7b**). This can be explained by the graph structure being more robust to adding random noise than the underlying distances. As most downstream single-cell analyses are based on graph representations rather than on distances directly, we believe that graph metrics thus more realistically represent data characteristics relevant for analysis. From the group of graph-based batch correction metrics, we decided not to use graph connectivity as it has a poor detection limit[2] and kBET due to computational costs[2], and from biological preservation metrics we excluded ARI, as it is computed in a very similar manner as NMI, leading to predominately redundant results (see scIB reproducibility code[3]), and cLISI, as we observed a poor detection limit with the metric being at its maximum in most cases, as also reported before[1].

While the NMI metric assesses the presence of cell clusters, which are of importance for any downstream single-cell analyses, its interpretation is complicated due to different factors that lead to poor correspondence between embedding clusters and cell type labels. Namely, NMI will be low both when we have merging of multiple cell types, usually caused by too high batch correction, as well as separation of cell types into multiple clusters, which can be caused by different factors. For example, in SATURN individual cell types separate into multiple clusters due to over-reliance on prior labels (**Supplementary Note S6**). In contrast, multiple clusters per cell type may also be a result of poor integration, leading to low NMI even when cell types separate well within batches. This is for example observed in GLUE with small alignment loss weight (0.005 or 0.0005, **Supplementary Figure S1**). Thus, in certain analyses where we wanted to strictly separate between cell type mixing, which is usually associated with excessive batch correction, and the ability to separate cells by cell types, indicating biological preservation within batches, we used an adapted NMI version (NMI-fixed). This metric uses a fixed clustering resolution and then annotates each cluster based on the majority ground-truth cell type. These annotations are used for NMI computation, thus indicating if cell types can be well separated. This corresponds to the ability to do standard scRNA-seq annotation where clustering at high resolution followed by cluster merging is often used to annotate populations that require high resolution to form distinct clusters. However, NMI-fixed can not detect over-clustering as observed in SATURN, thus we decided to use NMI in most other analyses as it is more generally able to detect if the integrated clusters directly correspond to cell types.

The previously suggested biological preservation metrics[1] are based mainly on cell type labels, which are commonly available only for cell types, but not for cell states within cell types. Some metrics, such as trajectory, cell cycle, or HVG conservation enable biological preservation evaluation at the sub-cell type level, however, they are often not applicable, due to the lack of a trajectory or cycling cells in the data and integration methods not producing corrected expression values, respectively. Thus, to evaluate how biological variation is preserved within cell types, we adapted the Moran's I biological preservation metric[4] that uses Moran's I to measure spatial covariation of gene expression across an embedding, corresponding to distinct gene expression patterns. Namely, we changed the metric to first identify variable genes within individual non-integrated samples and cell type groups and then

compare their variation before and after integration for every individual group. These metric modifications enable the selection of more relevant genes that are truly variable prior to integration, remove batch effect biases due to per-sample computation, and enable direct comparison before and after integration. We show examples of Moran's I values and the corresponding expression patterns for five gene groups known to be variable in mouse healthy adult beta cells[4] on embeddings produced with different integration models (**Supplementary Figure S7c**). However, as this metric is computed per sample, it can not detect the preservation of cross-sample patterns or proper alignment across samples. Sample alignment is much more challenging to measure as it usually requires prior knowledge about sample metadata, such as the presence of a developmental trajectory or disease-driven differences across samples within an individual cell type. Therefore, we show sample alignment on a few examples but do not propose a generic metric for it.

1.2. Supplementary Note S2: Effect of hyperparameter tuning on the performance of different models

One important characteristic of integration models is the ability to tune hyperparameters regulating batch correction to achieve sufficient integration for downstream analyses depending on the batch strength. However, we observed that some models did not offer enough flexibility in batch correction tuning and some hyperparameters weren't monotonically associated with batch correction, in contrast to our expectations. In the scVI and VAMP models, stronger KL regularization resulted in lower biological preservation, but iLISI did not increase consistently with increasing KL regularization strength and overall this did not lead to very high iLISI (**Supplementary Figure S1**, **Supplementary Figure S12**). This biological preservation-batch correction tradeoff makes this hyperparameter-model combination less favorable. In the GLUE and CYC models, some hyperparameters had consistently monotonic batch correction relationships. Multiple hyperparameters of GLUE (adversarial alignment loss weight, gene graph loss weight, and gene graph edge weight) affected integration performance in different ways (**Supplementary Figure S1**), complicating the selection of the hyperparameter set to tune. We observed that increased alignment loss weight, which should increase adversarial batch correction, had an optimal iLISI range (around 0.05 in all datasets), and thus did not increase batch correction beyond a certain point, similarly as described for KL regularization in the scVI and VAMP models. Note that the low NMI at low alignment loss weight is not caused by over-integration but rather by the separation of cells across the system, thus leading to multiple clusters per cell type, which then do not match up with the cell type labels (**Supplementary Note S1**). Furthermore, decreasing gene graph loss weight and gene graph weights led to higher iLISI, with low gene graph weights leading to the highest iLISI overall, at the expense of biological preservation (**Supplementary Figure S1**, **Supplementary Figure S12**). This could be explained by the role of the gene graph in connecting the gene embeddings and subsequently cell embeddings from individual encoders, thus supervising biological information in the integration. Furthermore, weaker graph constraints could lead to lower graph loss and lower graph loss results in relatively higher importance of the alignment loss during training. In the CYC the L_{CYC} weight was associated with increased batch correction at the expense of biological preservation. The addition of L_{CYC} to VAMP was also able to achieve higher iLISI than the tuning of KL regularization strength only, making L_{CYC} a better candidate for custom batch correction tuning of VAMP than KL regularization strength. Lastly, as SATURN didn't offer many loss parameters to be optimized, we tuned only protein similarity loss weight that regulates the reliance on prior-defined protein embedding of genes. Despite a wide range of the tested values (0.01-10.0), we did not observe major variations in biological preservation or batch correction. It is important to note that observed hyperparameter-batch correction patterns across models hold only in the tested hyperparameter ranges. Nevertheless, the tested hyperparameter ranges were selected so that values outside of them would likely lead to inadequate biological preservation or batch correction, making them unsuitable for integration. Overall, our results show the complexity of batch correction tuning.

1.3. Supplementary Note S3: KL regularization strength tuning leads to dimension shrinkage, but not batch correction

Increasing KL regularization strength led, as expected, to stronger batch correction and lower biological preservation. However, we observed that this was not associated with the indistinguishability of cell representations in the latent space, but rather was caused by a reduced number of latent dimensions effectively used in downstream cell graph computation. Namely, stronger KL regularization leads to more latent dimensions having variation near zero (**Supplementary Figure S4**). This occurs as the cell embedding is the predicted mean of

the cell's latent variable representation, as is also done in scVI and GLUE, which is pushed towards zero to match the Gaussian prior when increasing KL regularization. As neighbors graph computation is standardly done with Euclidean distance on non-scaled embedding[5] this effectively leads to a small number of latent dimensions having a stronger effect on the graph representation. Indeed, after applying standard scaling to the embedding prior to neighbors computation the different KL regularization strengths no longer strongly affected NMI and iLISI (**Figure 3**, **Supplementary Figure S13**, **Supplementary Figure S6**). This indicates that biological and batch information is preserved in the embedding, but is not captured by the standard graph computation protocol. Nevertheless, we observed a drop of within-cell type variation (Moran's I) on both scaled and unscaled data when increasing KL regularization strength, suggesting that finer variation is nevertheless lost. In contrast, while increasing L_{CYC} loss weight also led to the shrinkage of latent dimensions, the embedding was not sensitive to scaling. We achieved this by computing cycle distances on latent embedding standardized within a minibatch. Thus, while KL regularization strength tuning in standard cVAE merely leads to a lower number of used latent dimensions, thus reducing both biological and batch information at the same time, L_{CYC} actually removes batch effects from the embedding, as indicated by higher batch correction at comparable biological preservation (**Supplementary Figure S1**).

1.4. Supplementary Note S4: Adversarial loss leads to the mixing of imbalanced cell types and results in the loss of fine biological variation

As we increased the L_{ADV} loss weight of GLUE or other parameters leading to higher iLISI, we observed mixing of unrelated cell types that could be easily distinguished by other models, such as delta, acinar, and immune cells in the mouse-human pancreatic dataset; amacrine and retinal ganglion cells in the organoid-tissue retinal dataset; and adipocyte and ASPC cells in cell-nuclei adipose dataset (**Figure 2b**, **Supplementary Figure S2**). Mixing usually included one cell type that was underrepresented in one of the systems (**Supplementary Figure S8**), thus requiring mixing with a cell type present in the second system for improving L_{ADV}. In contrast, L_{CYC} was not sensitive to such system imbalance. This can be explained by L_{ADV} being in principle cell type agnostic, namely, all cells are jointly classified based on their system, while L_{CYC} directly compares a cell and its representation in another system, thus never comparing cells with different biological backgrounds (**Figure 1d**). GLUE aims to combat this by down-weighting the contribution of cells from unbalanced clusters to L_{ADV}, with clusters being obtained from an initial coarse integration round. Nevertheless, as our results show, this is not sufficient to prevent mixing in practice. The observed mixing may also explain why GLUE has lower Moran's I scores, namely it may mix cells within cell types when there are differences in cell state proportions across systems, similarly as for cell types.

1.5. Supplementary Note S5: Excessive reliance on batch covariate information in scGEN leads to over-integration

The scGEN model first trains a normal cVAE and then applies an additional batch correction step to every cell type separately. This is done by selecting one batch as the reference and moving all other batches (queries) on top of the reference batch. First, the distance between the reference and query mean latent embedding is computed and then query cell embeddings are transformed by the addition of the distance vector. This will force batch overlap even in the presence of biological differences. For example, all models except scGEN correctly separated retinal pigment epithelial cells from organoid and primary tissue data[6] (**Supplementary Figure S2a**). In the Mueller cell example, where organoid cells are expected to align with the tissue periphery but not fovea cells[6], scGEN with samples as batch covariates led to over-integration, placing all samples on top of each other (**Supplementary Figure S3**). The alignment was also not improved by using systems rather than samples as the batch covariate, which instead led to under-integration on the sample level within systems. This also resulted in poor alignment of systems as the alignment with latent space arithmetic was disrupted due to the per-system embedding structure and its mean being strongly affected by sample batch composition rather than being dominated by biology (**Supplementary Figure S2a, Supplementary Figure S3**). This highlights another issue, namely that there is no guarantee that the alignment will be correct on the level of cell subtypes for which corresponding cell annotation is unavailable. This is a consequence of the model never being faced with the task of reconstructing cell expression from the final corrected latent space. For example, we observed some errors in the amacrine subtype alignment, where cells expressing starburst amacrine marker *SLC18A3* did not co-localize in the scGEN embedding while showing clear sub-localization when using the VAMP+CYC model (**Supplementary Figure S14**).

Therefore, we deem scGEN inappropriate for integration as its latent space is inherently biologically flawed due to over-reliance on prior information about cell types and batch covariates.

1.6. Supplementary Note S6: Excessive reliance on prior cell cluster labels in SATURN harms cell representation learning

SATURN uses prior cell cluster information to guide cell representation fine-tuning of a pre-trained conditional autoencoder (cAE)-based embedding. This is achieved by maximizing within-species distances between cells with different labels and minimizing cross-species distances between cells that are likely to have the same label. This leads to good cell type level preservation when high-quality cell type labels are available per species (SATURN-CT), however, it does not capture well within cell type information (**Figure 2a**). This can be explained by the lack of cell expression representation objective in the fine-tuning step, which relies on labels alone. Furthermore, while the authors propose that clusters can be used as a source of prior label information, we observed that this led to poor NMI (**Figure 2a**). Namely, when multiple batches are present within species, as the datasets in our mouse data, this will result in multiple dataset-specific clusters for individual cell types, which will also be reflected in the separation of cells by prior cluster in the final integration (**Supplementary Figure S15**). Similar effects were also observed when prior clusters were too fine or coarse with respect to underlying biological variation. For example, immune cells were separated into three distinct clusters when using cluster-based prior (SATURN), directly corresponding to prior clusters, and were merged into a single cluster when using cell-type-based prior (SATURN-CT). In both cases, the finer cell cluster structure retained within other models was lost (**Figure 2b**).

1.7. Supplementary Note S7: Learning of prior components in VAMP

The VampPrior, as proposed in the original publication[7], contains learnable components and during training the components evolve to be located within data-dense regions (**Supplementary Figure S16**). Interestingly, if prior components are fixed, this does not affect VAMP integration performance, but strongly affects GMM where batch correction drops to a similar level as when using a single prior component (**Supplementary Figure S10**). This is somewhat unexpected, as one would assume that the encoder would be better able to align input representations to fixed prior components when inputs and prior parameters do not lie in the same space, as is the case in the GMM, but not the VAMP.

We also inspected the effect of prior component initialization. VAMP pseudoinputs were initialized with input data. We did not use random initialization as it is challenging to perform appropriate random initialization of pseudoinputs in the input space that would match the input cell distribution and could thus be effectively encoded by the encoder, which is trained to generate input cell representations. In the GMM, the data-based and random prior initialization showed no clear differences (**Figure 4**). Furthermore, the choice of input cells used to initialize pseudoinputs in VAMP did not seem to play a role, with similar performance when initializing them from a single cell type or system or in a balanced fashion from all cell types or systems (**Supplementary Figure S11**). This indicates robustness against varying prior initialization.

2. Supplementary methods

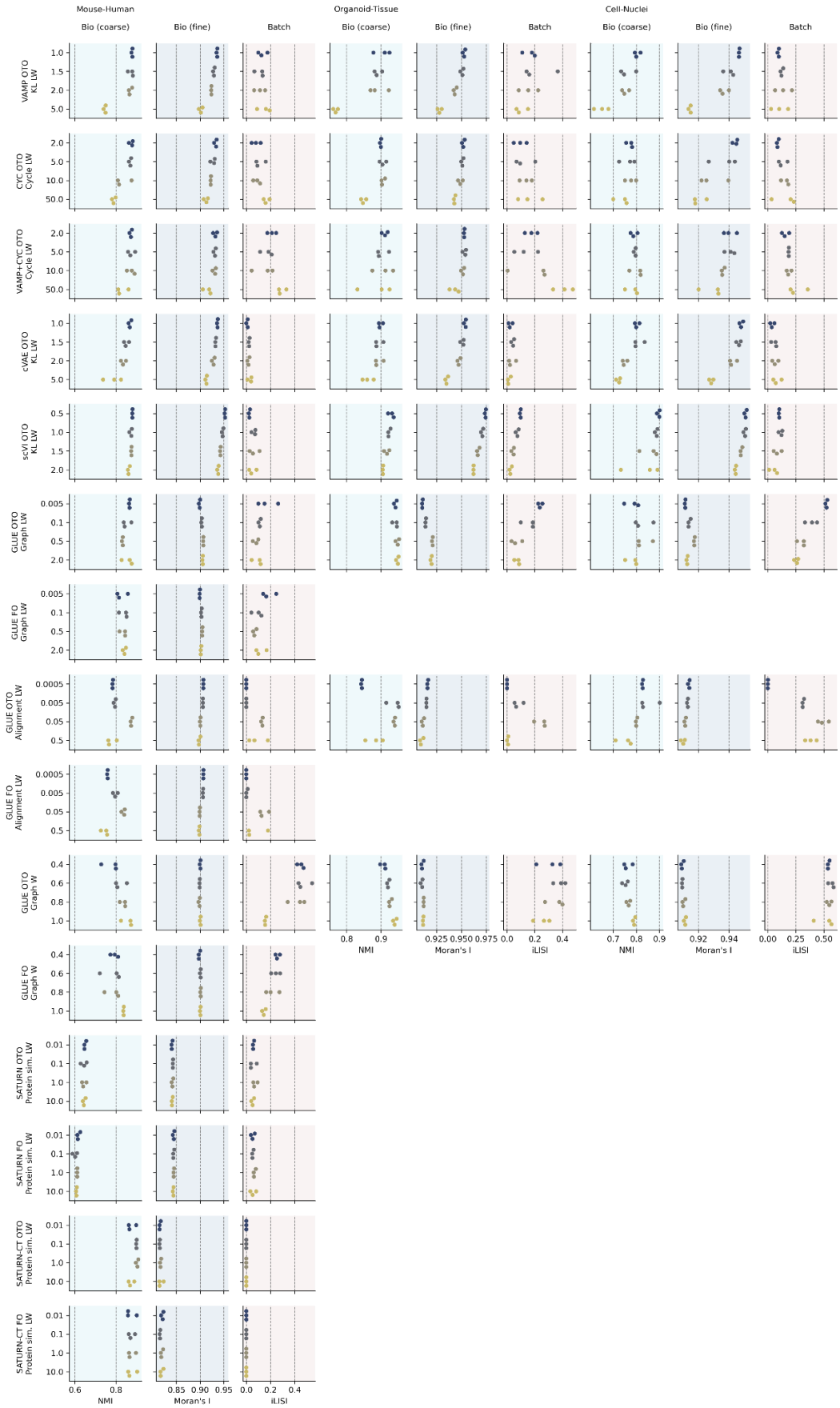
2.1. Simulation for comparison of graph and distance-based metrics

We simulated two groups of points (n=100 per group) from normal distributions with means of zero and one, respectively, and variance of one. We added 15 additional dimensions coming from a standard normal distribution, representing noise dimensions that do not separate between the two groups. ASW and LISI (both biological preservation and batch correction variants) were computed on all 16 dimensions while the size of 0 to 15 noise dimensions was divided by ten to push them towards zero. The simulations were repeated ten times.

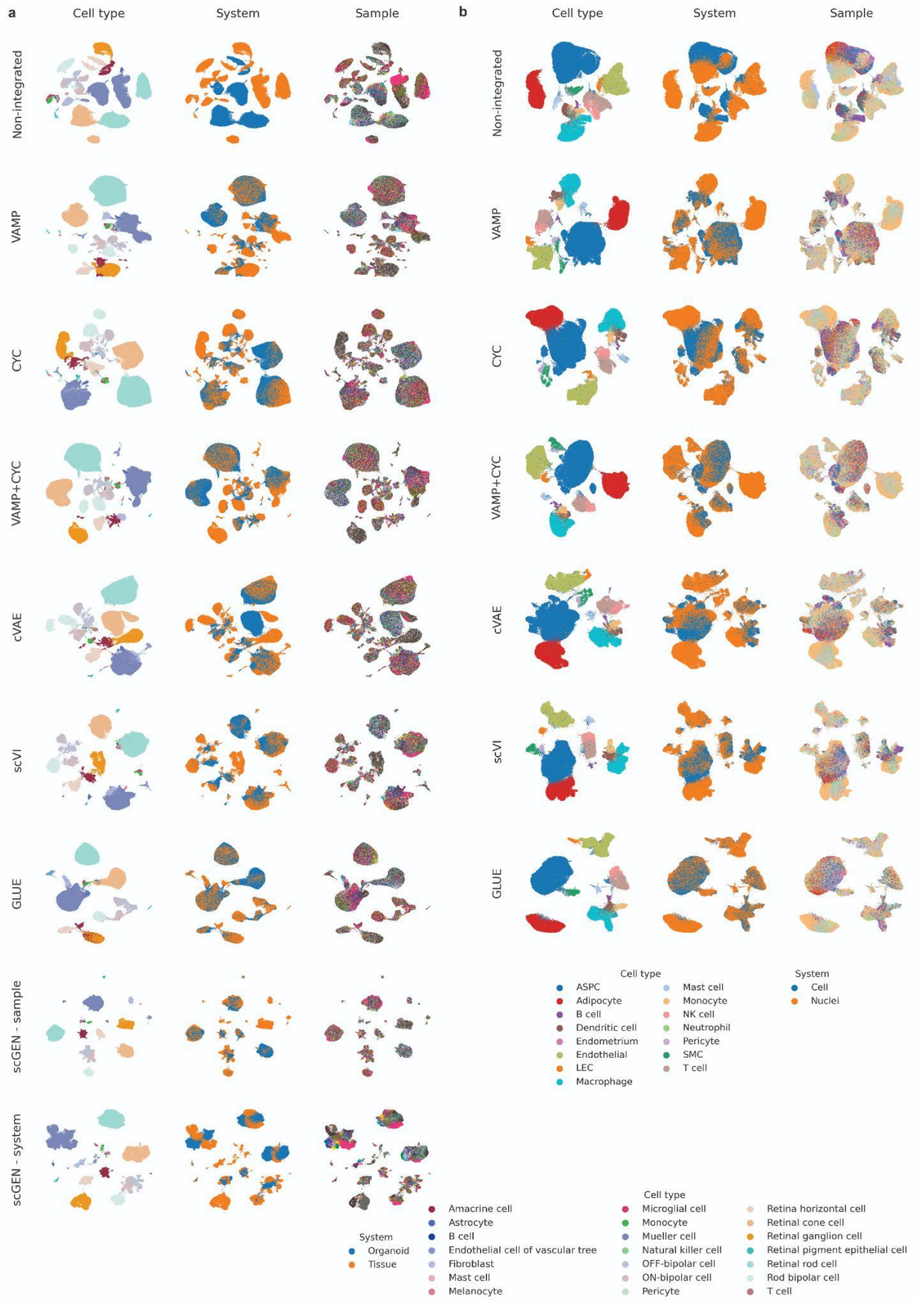
2.2. Integration with scGEN

We ran scGEN with default parameters, except for *kl_weight=0.1*. We used either samples or systems as the batch variable since the model does not allow for multiple batch variables.

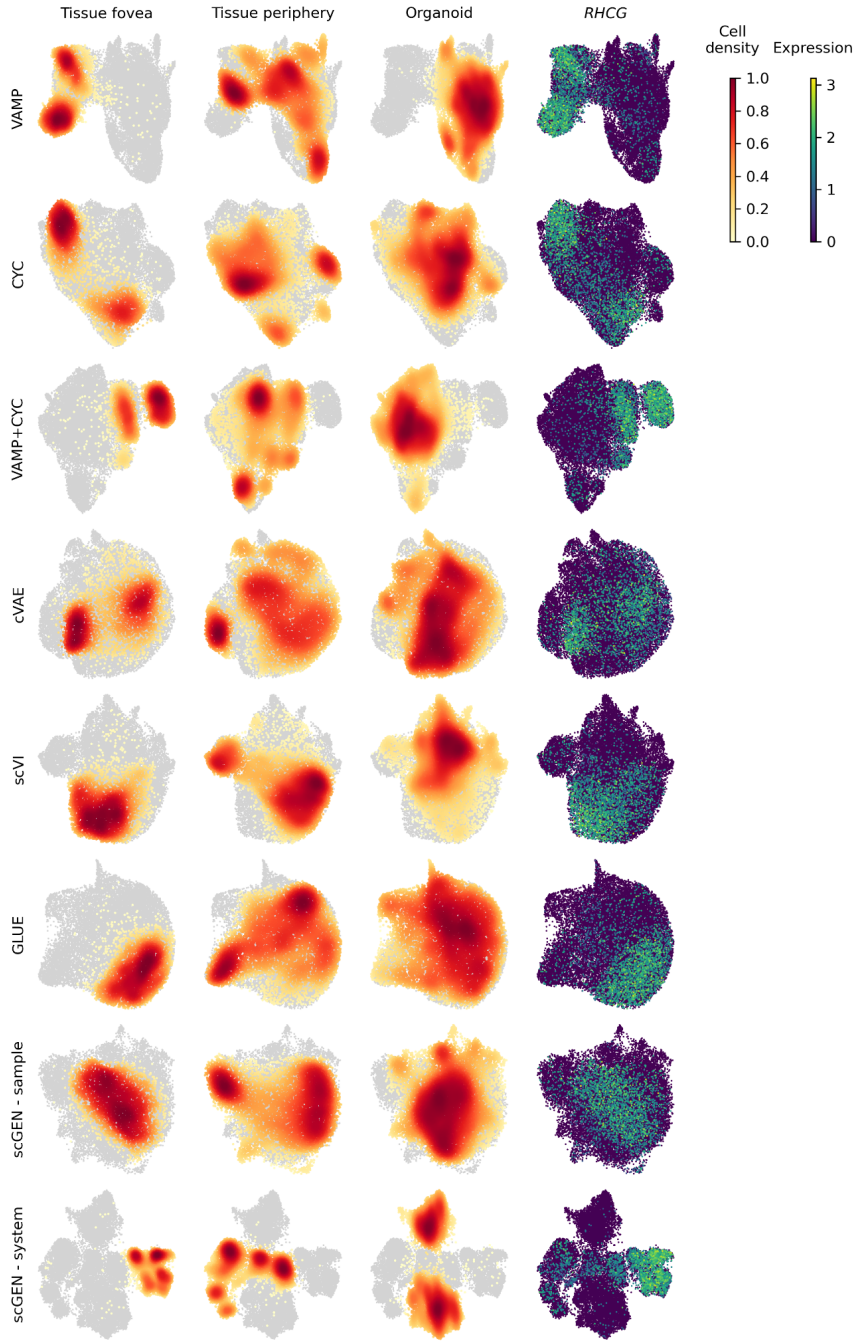
3. Supplementary figures


Mouse-Human
Organoid-Tissue
Cell-Nuclei
Bio (coarse)
Bio (fine)
Batch
VAMP OTO KL LW
CYC OTO Cycle LW
VAMP+CYC OTO Cycle LW
cVAE OTO KL LW
scVI OTO KL LW
GLUE OTO Graph LW
GLUE FO Graph LW
GLUE OTO Alignment LW
GLUE FO Alignment LW
GLUE OTO Graph W
GLUE FO Graph W
SATURN OTO Protein sim. LW
SATURN FO Protein sim. LW
SATURN-CT OTO Protein sim. LW
SATURN-CT FO Protein sim. LW
NMI
Moran's I
iLISI

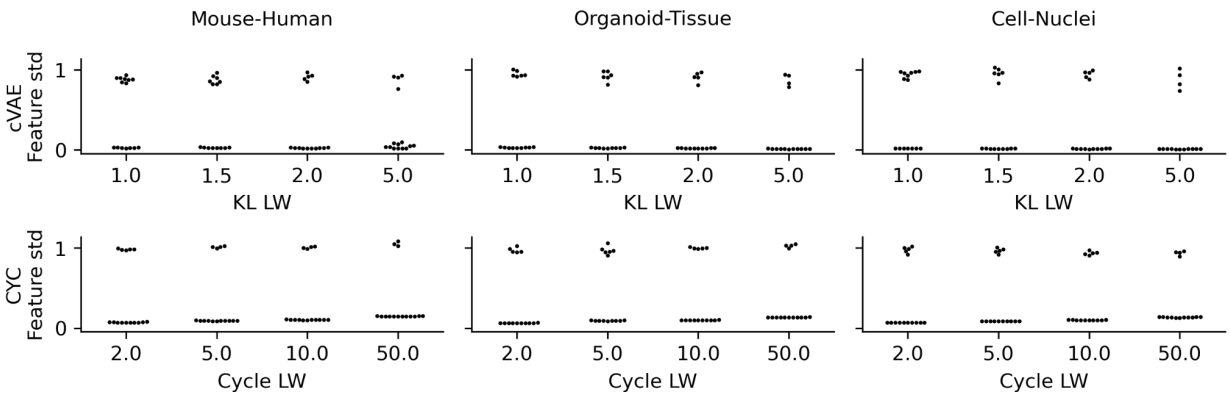
Supplementary Figure S1: Integration performance across models and different model hyperparameters. Rows show different models and tested hyperparameters (y-axis values), columns show per-dataset integration metrics, and dots represent individual runs with different seeds.


a
Cell type
System
Sample
b
Cell type
System
Sample
Non-integrated
VAMP
CYC
VAMP+CYC
cVAE
scVI
GLUE
scGEN - sample
scGEN - system
Cell type
ASPC
Adipocyte
B cell
Dendritic cell
Endometrium
Endothelial
LEC
Macrophage
Mast cell
Monocyte
NK cell
Neutrophil
Pericyte
SMC
T cell
System
Cell
Nuclei
System
Organoid
Tissue
Amacrine cell
Astrocyte
B cell
Endothelial cell of vascular tree
Fibroblast
Mast cell
Melanocyte
Cell type
Microglial cell
Monocyte
Mueller cell
Natural killer cell
OFF-bipolar cell
ON-bipolar cell
Pericyte
Retina horizontal cell
Retinal cone cell
Retinal ganglion cell
Retinal pigment epithelial cell
Retinal rod cell
Rod bipolar cell
T cell

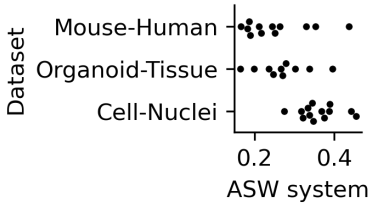
Supplementary Figure S2: UMAP embeddings of representative runs for the best hyperparameter setting of every model. Shown for datasets (**a**) organoid-tissue and (**b**) cell-nuclei. To (**a**) we also added two example runs for scGEN with samples or systems as batch covariates. A legend for sample colors is not shown due to the large number of samples.



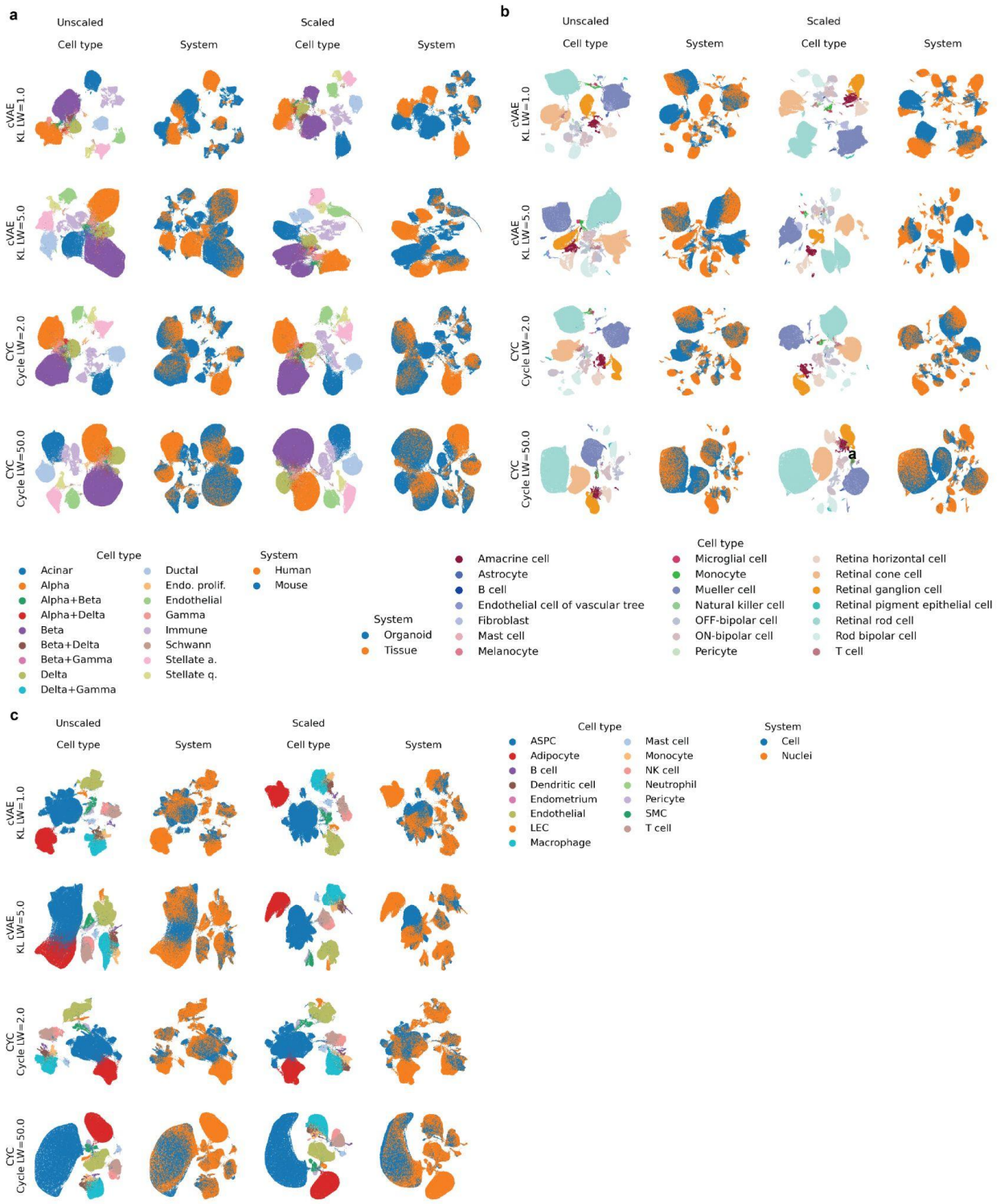
Supplementary Figure S3: Sample alignment in integrated Mueller cells. Plotted are UMAPs of Mueller cell subset from the organoid-tissue dataset based on representative runs for the best hyperparameter setting of every model. We also added two example runs for scGEN with samples or systems as batch covariates. The UMAPs are colored by cell density in organoid and tissue cell groups and expression of foveal marker *RHCG*.



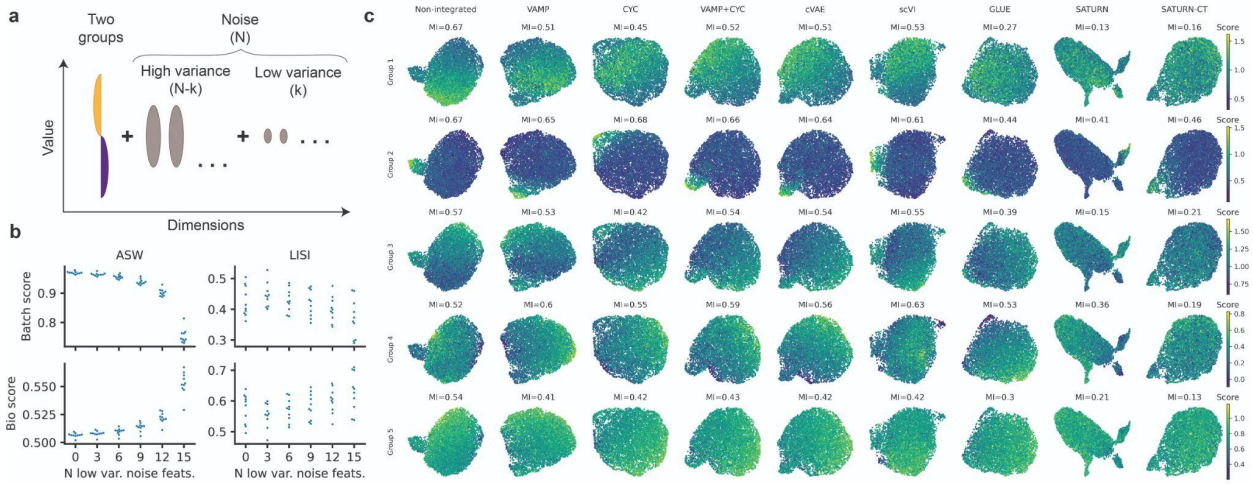
Supplementary Figure S4: Shrinkage of latent embedding dimensions when increasing batch correction-related hyperparameters. Standard deviations (std) of individual embedding dimensions (dots) obtained with different hyperparameter values (x-axis, shown for runs with seed=1) for cVAE and as a comparison CYC (rows) across datasets (columns).



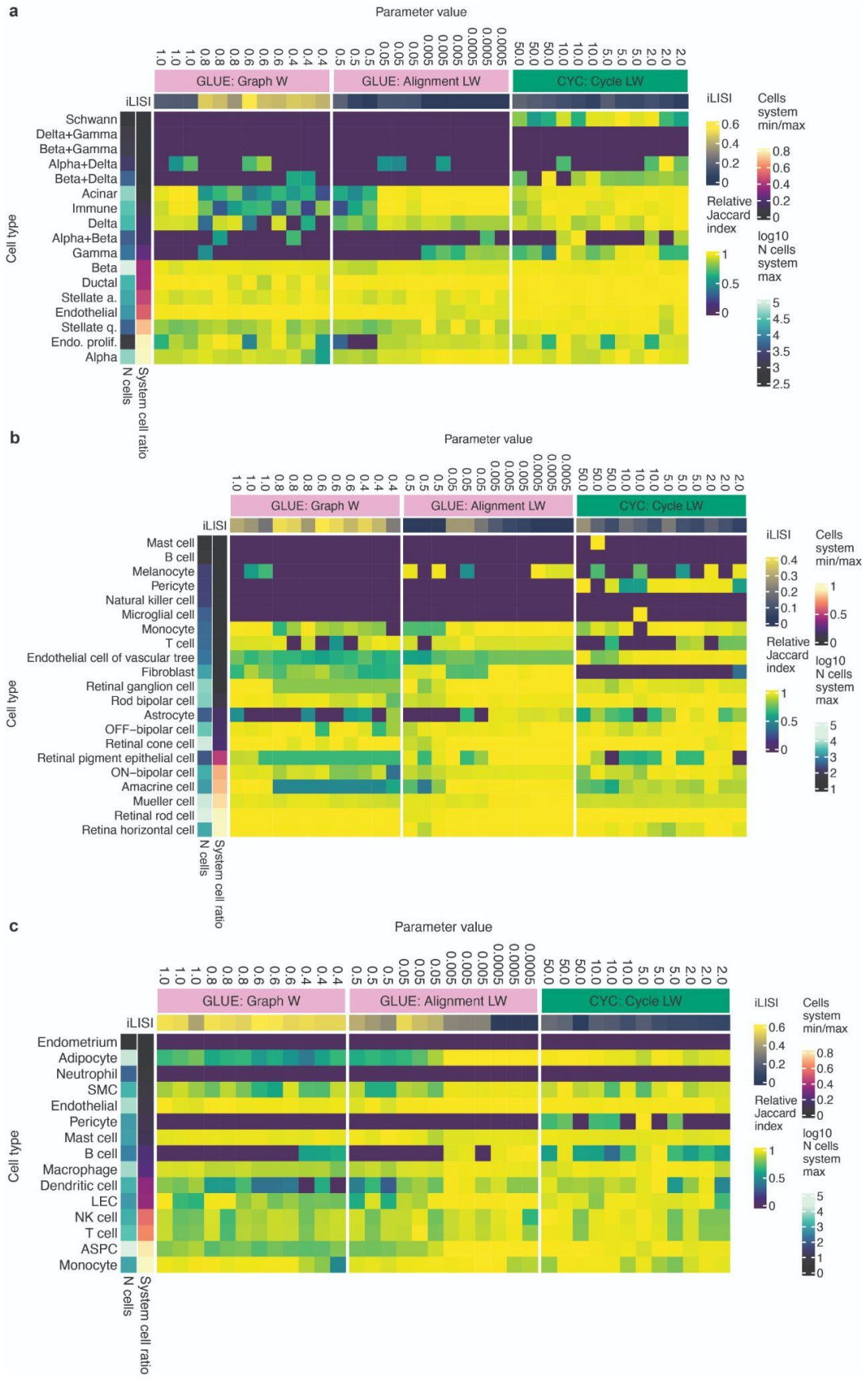
Supplementary Figure S5: Between-system batch effects strength prior to integration. Shown are ASW batch values computed with systems as batch covariates for individual cell types (points) in every dataset, with higher score indicating stronger system mixing. The distribution of ASW batch scores is significantly higher in cell-nuclei dataset compared to mouse-human (p-value = 2.1e-03) or organoid-tissue (p-value = 3.9e-03) dataset, with the latter two not differing significantly from each other (p-value = 3.7e-01). This indicates that batch effects in the cell-nuclei dataset are relatively weaker, as can also be seen in the non-integrated UMAP plots (**Figure 2b**, **Supplementary Figure S2**).



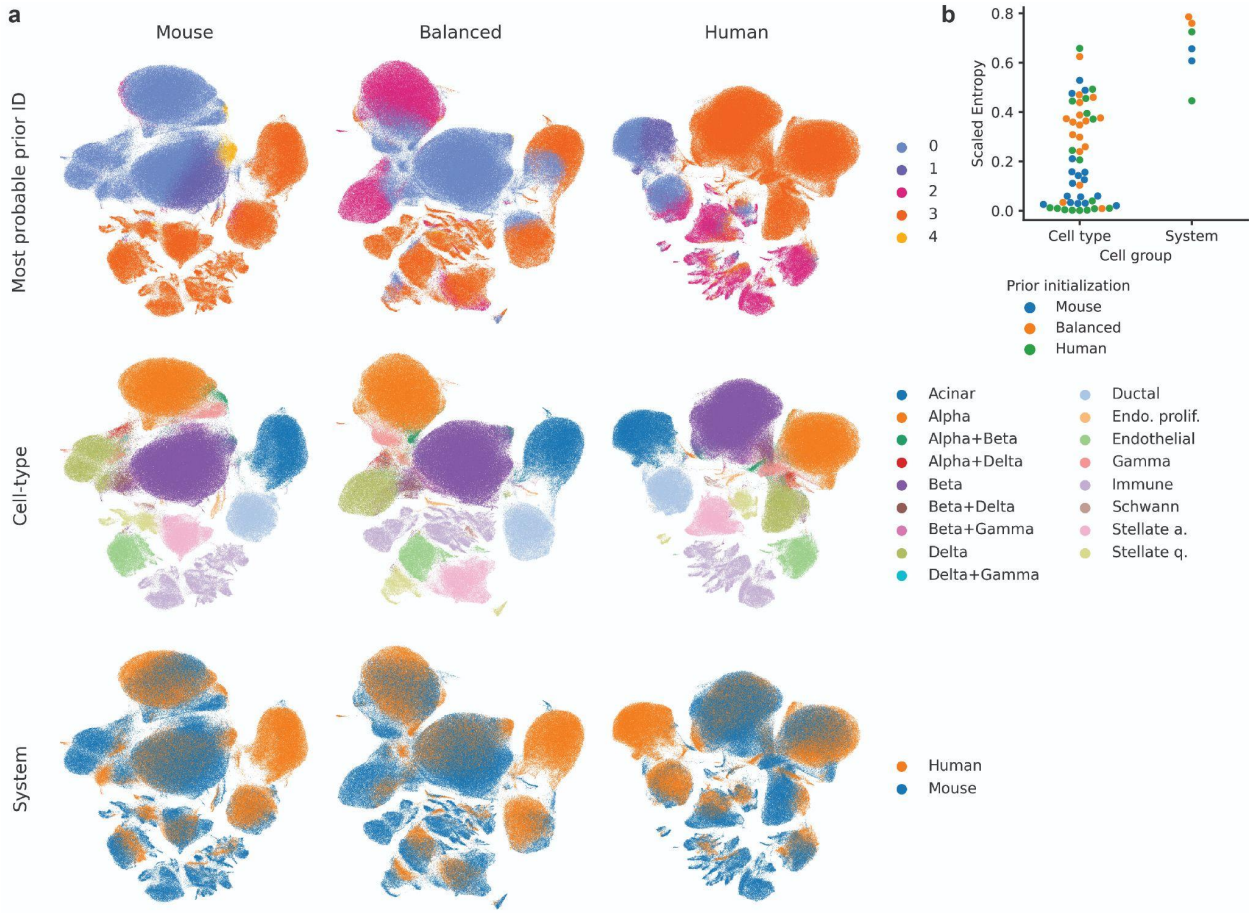
Supplementary Figure S6: UMAP of scaled and unscaled embeddings. Embeddings produced with cVAE and CYC models with low and high batch correction strength hyperparameter values (shown runs with seed=1). Datasets: (**a**) mouse-human, (**b**) organoid-tissue, (**c**) cell-nuclei.



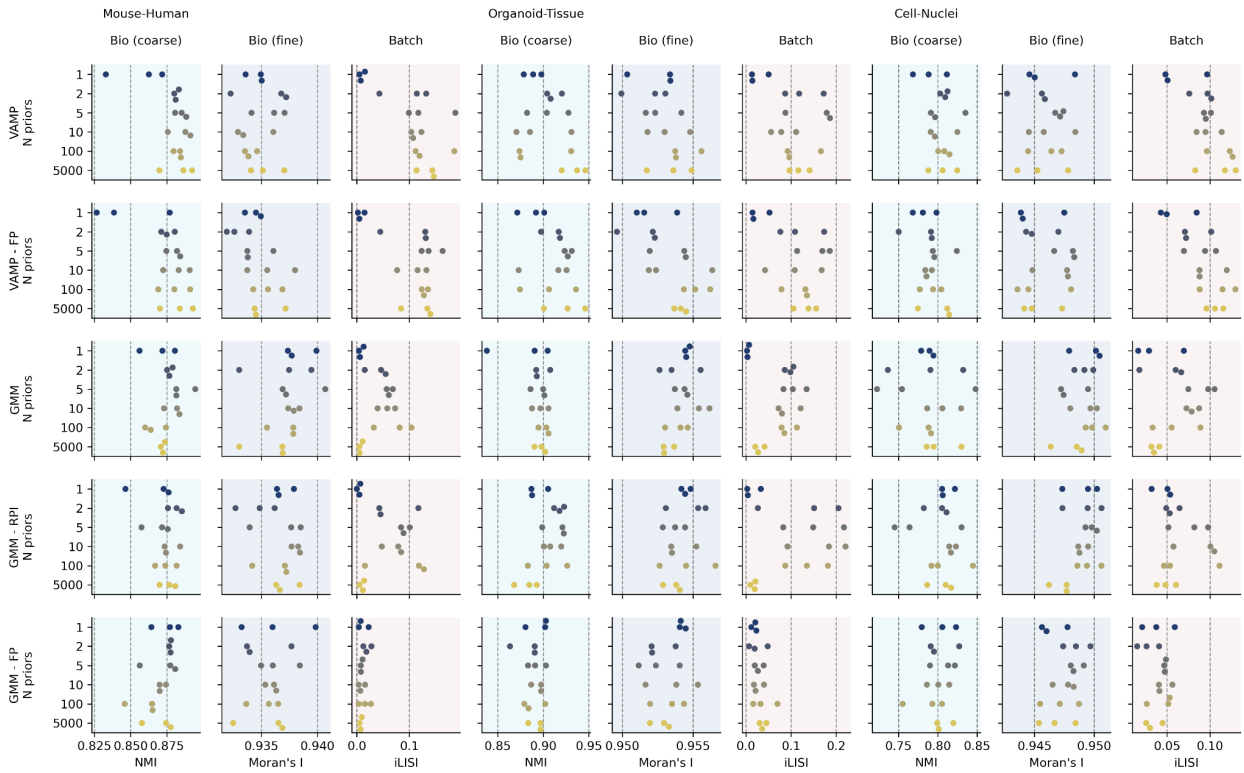
Supplementary Figure S7: Evaluation of integration metrics. (**a**) Overview of the data simulation process used for comparing distance and graph-based metrics. (**b**) Graph-based metrics are less affected by the value range of the noise dimensions than distance-based metrics. Shown are biological preservation and batch correction scores based on ASW or LISI when 0 to 15 of the noise dimensions are shrunk to have smaller variance. (**c**) Examples of Moran's I scores and corresponding expression patterns on UMAPs for five gene groups known to be variable in healthy adult mice. For every model, one representative run from the best hyperparameter setting was selected (as shown in **Figure 2b**) and cells were subsetted to one healthy adult mouse pancreatic sample, used for computing UMAP and Moran's I.


a
Parameter value
GLUE: Graph W
GLUE: Alignment LW
CYC: Cycle LW
iLISI
Cell type
Schwann
Delta+Gamma
Beta+Gamma
Alpha+Delta
Beta+Delta
Acinar
Immune
Delta
Alpha+Beta
Gamma
Beta
Ductal
Stellate a.
Endothelial
Stellate q.
Endo. prolif.
Alpha
N cells
System cell ratio
Cells system min/max
Relative Jaccard index
log10 N cells system max
b
Mast cell
B cell
Melanocyte
Pericyte
Natural killer cell
Microglial cell
Monocyte
T cell
Endothelial cell of vascular tree
Fibroblast
Retinal ganglion cell
Rod bipolar cell
Astrocyte
OFF−bipolar cell
Retinal cone cell
Retinal pigment epithelial cell
ON−bipolar cell
Amacrine cell
Mueller cell
Retinal rod cell
Retina horizontal cell
c
Endometrium
Adipocyte
Neutrophil
SMC
Endothelial
Pericyte
Mast cell
B cell
Macrophage
Dendritic cell
LEC
NK cell
T cell
ASPC
Monocyte

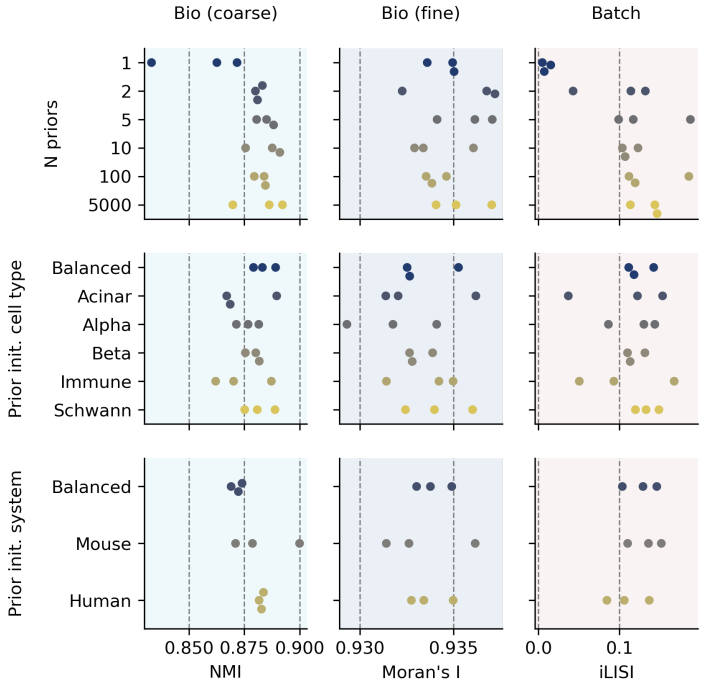
Supplementary Figure S8: Per-cell type evaluation of cell type level biological preservation. Shown are all runs of selected models and hyperparameters, alongside with batch correction score (x-axis). Individual cell types are annotated with the number of cells in the more abundant system and the ratio of cells between the less and the more abundant system (y-axis). Datasets: (**a**) mouse-human, (**b**) organoid-tissue, (**c**) cell-nuclei.



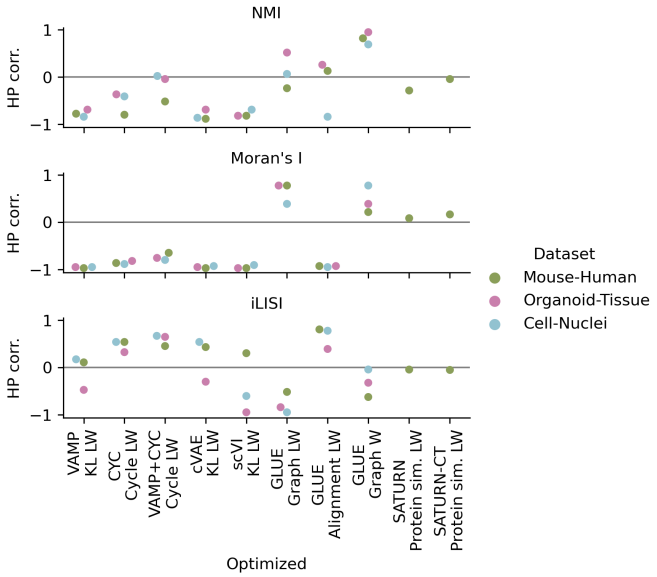
Supplementary Figure S9: Individual VampPrior prior components co-localize with cell types rather than systems. (**a**) Cells are colored by the prior component (N=5) with the highest probability, cell type, and system (rows) on embeddings generated with different prior initializations (columns). Priors were initialized either by sampling cells from a single system or in a balanced manner from both systems. Shown are UMAPs of the VAMP integration of the mouse-human dataset. (**b**) Entropy of the most probable prior assignment within every cell type or system (dots) corresponding to (**a**), with colors indicating runs with different prior initializations. Entropy was scaled by the maximal possible entropy.



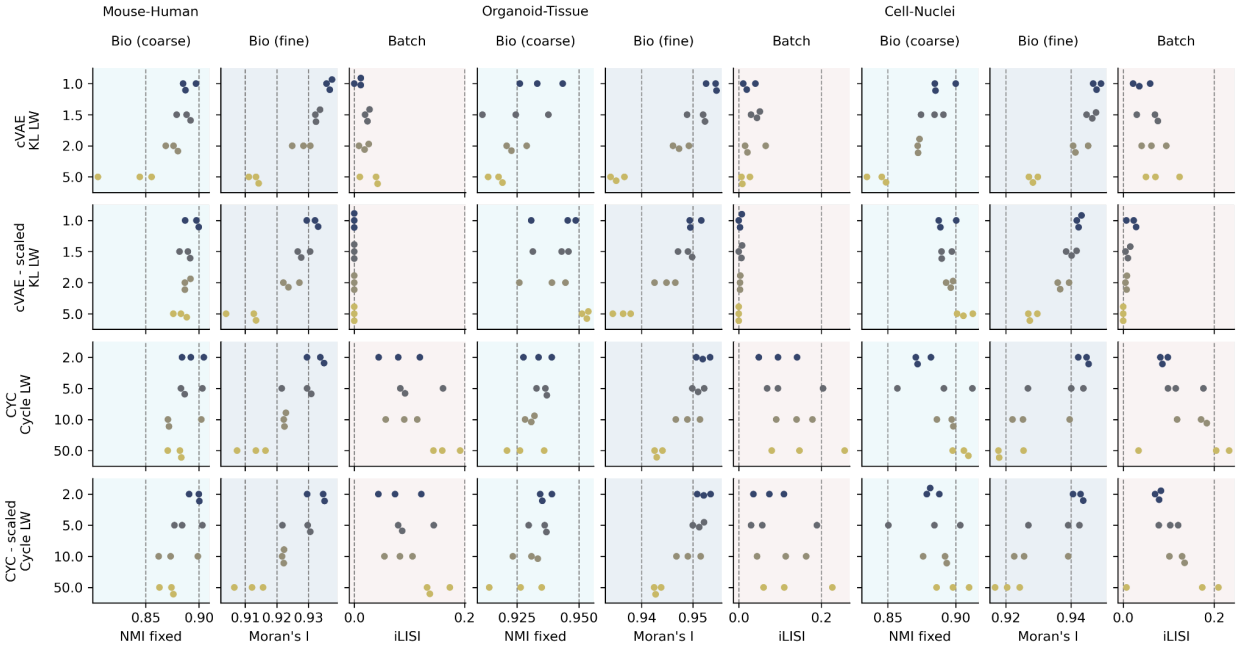
Supplementary Figure S10: Integration performance of VAMP and GMM with different prior component settings. Rows show different models and prior settings, including the number of prior components (y-axis values), columns show per-dataset integration metrics, and dots represent individual runs with different seeds for each setting. FP - fixed prior, RPI - random prior initialization.



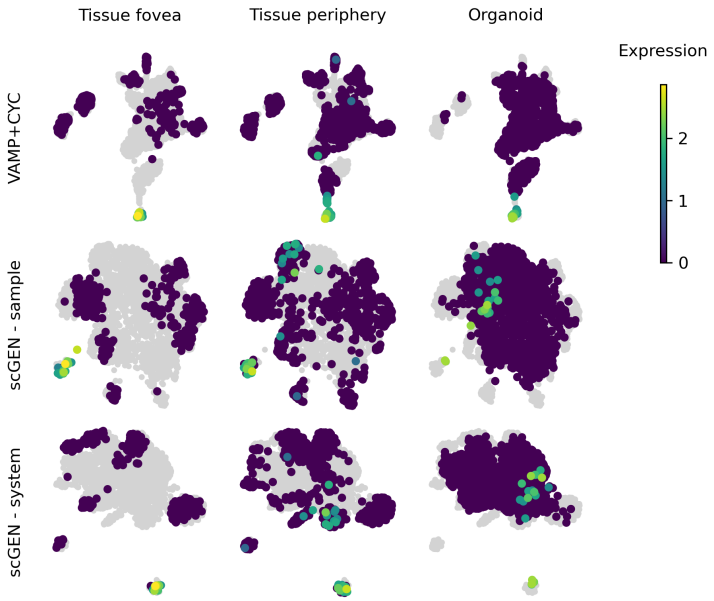
Supplementary Figure S11: Initializing pseudoinputs from all cell types or both systems or only one cell type or system does not have a major effect on the integration performance. Rows show different prior initializations, with the top row having random initialization with a different number of prior components (y-axis), and the other two rows being initialized with either one prior component sampled from each cell type or system ("Balanced") or from a single cell type or system with the number of prior components equal to the number of cell types (N=17) or systems (N=2), respectively. Columns show integration metrics for the mouse-human dataset and dots represent individual runs with different seeds.



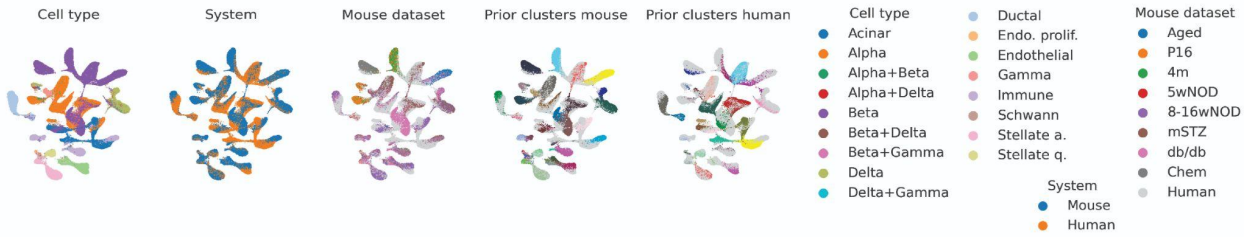
Supplementary Figure S12: Spearman correlation between hyperparameter values and integration metrics. Correlation (corr.) was computed for different combinations of models and tuned hyperparameters (x-axis) across all runs and hyperparameter values (HP) in OTO setting as shown in **Supplementary Figure S1**, separately for each dataset.



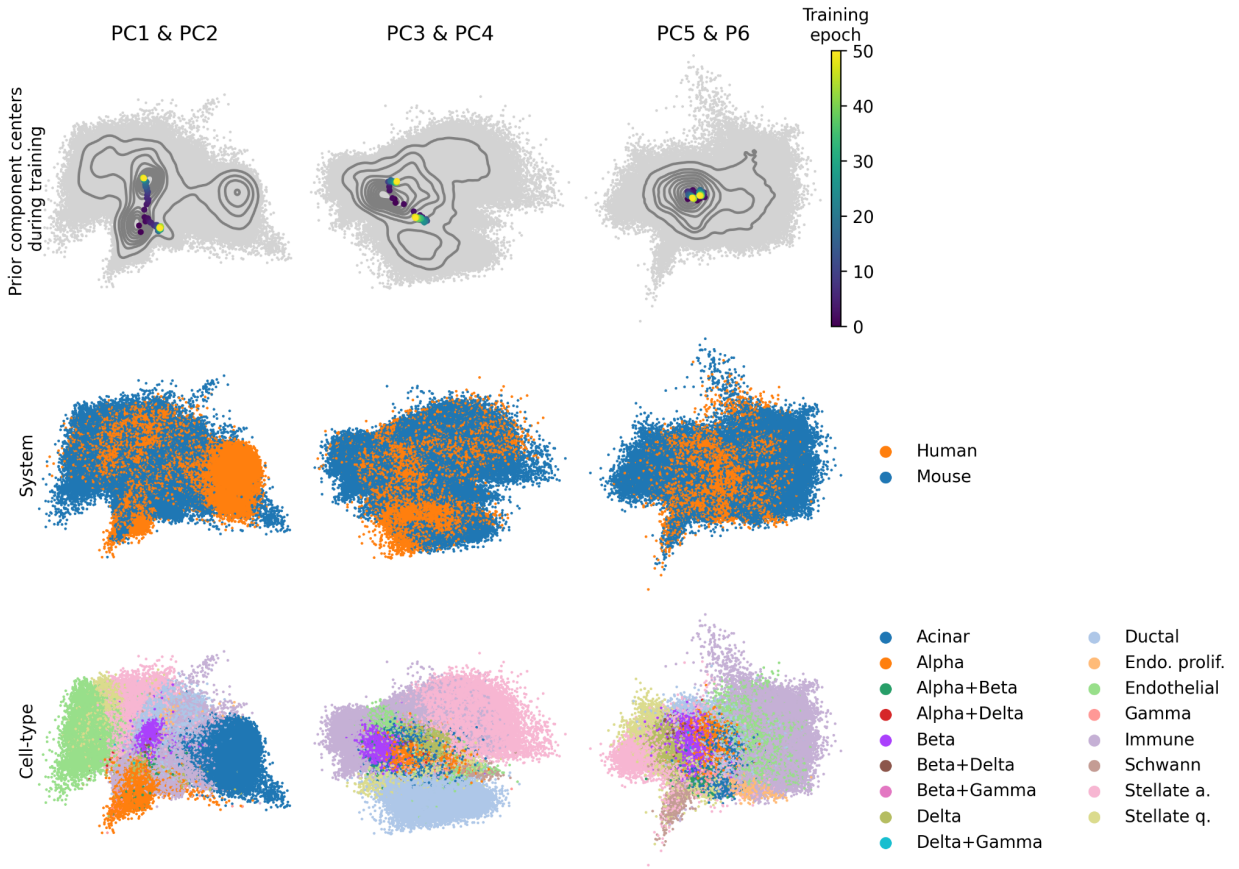
Supplementary Figure S13: Effect of scaling cVAE integration embedding on integration metrics. Integration metrics of datasets (x-axis) computed using scaled and unscaled embedding from cVAE and for comparison CYC, shown across different values of hyperparameters used for tuning batch correction (y-axis).



Supplementary Figure S14: Cell subtype preservation in amacrine cells after integration. The UMAPs are colored by expression of starburst amacrine cell marker *SLC18A3*. They were computed for the amacrine cell subset from the organoid-tissue dataset based on a representative run for the best hyperparameter setting of VAMP+CYC and on two example runs of scGEN with samples or systems as batch covariates.



Supplementary Figure S15: Batch effects in prior cluster labels are reflected in the final SATURN integration. The UMAPs display the per-system prior clusters and the dataset covariate for the mouse system. They were computed on a representative run from the best hyperparameter setting for SATUR mouse-human integration. Prior cluster color legend is not shown as individual cluster names are not meaningful. For the mouse-specific covariates (datasets and clusters) the human data is shown as a background in light gray, and vice versa for the human covariates.



Supplementary Figure S16: The update of the VampPrior components during the training. The top row shows how the VampPrior components' (N=2) mean parameters (colored) are updated during the training in the space of the first six PCs (pairwise plots in columns) computed on the model's latent space. The contour lines represent the density of the input cells (every 10th quantile) and the gray points individual cells. The middle row shows the system covariate and the bottom row shows the cell type. Shown are UMAPs of VAMP integration of the mouse-human dataset.

4. Supplementary tables

Supplementary Table S1: Comparison of pre-integration sample distances within and between systems. One-sided Mann–Whitney U test was used to compare distributions of per-cell-type sample distances in the PCA space within a system (control) and between systems (case) for every data setting (sheet names). Column names: cell_type - tested cell type; system - control group, with "within/between" in the mouse-human data setting referring to within/between dataset distances in the mouse data; u - test statistic; padj - adjusted p-value; n_system and n_corssystem - number of distances (observations) in the control and case groups, respectively.

Accessible at: https://drive.google.com/drive/folders/1UCrhSr94cJdROy04l4m56GqNz1bgclkV?usp=share_link

References

1. Luecken, M. D. *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).
2. Lütge, A. *et al.* CellMixS: quantifying and visualizing batch effects in single-cell RNA-seq data. *Life Sci Alliance* **4**, (2021).
3. Metrics correlation and range analysis. *GitHub* https://github.com/theislab/scib-reproducibility/blob/main/notebooks/analysis/Metrics_correlation_and_range_analysis.ipynb.
4. Hrovatin, K. *et al.* Delineating mouse β-cell identity during lifetime and in diabetes with a single cell atlas. *bioRxiv* 2022.12.22.521557 (2022) doi:10.1101/2022.12.22.521557.
5. Heumos, L. *et al.* Best practices for single-cell analysis across modalities. *Nat. Rev. Genet.* 1–23 (2023).
6. Cowan, C. S. *et al.* Cell Types of the Human Retina and Its Organoids at Single-Cell Resolution. *Cell* **182**, 1623–1640.e34 (2020).
7. Tomczak, J. M. & Welling, M. VAE with a VampPrior. *arXiv [cs.LG]* (2017).