ARTICLES
https://doi.org/10.1038/s42256-022-00443-1
nature machine intelligence

Check for updates

OPEN

Simultaneous dimensionality reduction and integration for single-cell ATAC-seq data using deep learning

Wolfgang Kopp[1,3] ✉, Altuna Akalin[1] and Uwe Ohler[1,2] ✉

Advances in single-cell technologies enable the routine interrogation of chromatin accessibility for tens of thousands of single cells, elucidating gene regulatory processes at an unprecedented resolution. Meanwhile, size, sparsity and high dimensionality of the resulting data continue to pose challenges for its computational analysis, and specifically the integration of data from different sources. We have developed a dedicated computational approach: a variational auto-encoder using a noise model specifically designed for single-cell ATAC-seq (assay for transposase-accessible chromatin with high-throughput sequencing) data, which facilitates simultaneous dimensionality reduction and batch correction via an adversarial learning strategy. We showcase its benefits for detailed cell-type characterization on individual real and simulated datasets as well as for integrating multiple complex datasets.

Rapid advances in single-cell epigenomics technologies, including single-cell assay for transposase-accessible chromatin with high-throughput sequencing (scATAC-seq), have enabled the interrogation of gene regulation at an unprecedented resolution. scATAC profiles the accessibility of chromatin across the whole genome and is currently the most widely adapted protocol to identify candidates of regulatory regions of importance to the system under investigation. The resulting datasets require specialized computational tools to cope with their characteristic high dimensionality and sparsity and will rely on scalability for future datasets.

A key step in every scATAC-seq processing pipeline is dimensionality reduction, which aims to represent the most salient trends in the data at lower dimensionality, such as groups of similar cells. The quality of this step is critical, as it precedes other analysis tasks, including cell-type characterization, identifying cell-type specific regulatory regions, motif analysis and so on. Several methods have been introduced for dimensionality reduction using scATAC-seq data, including scABC[1], chromVAR[2], Scasat[3], latent Dirichlet allocation (cisTopic)[4], latent semantic indexing (LSI)[5], iterative LSI[6], SnapATAC[7], SCALE[8], scDEC[9] and PeakVI[10]. A recent benchmark analysis showed that current computational tools work well for cell-type characterization for small or moderate dataset sizes, but may not scale to large dataset sizes and/or vary in performance across different datasets[11]. Apart from performing dimensionality reduction, the growing number of published datasets opens up new avenues for data integration of replicates or data obtained with different protocols, such as combinatorial indexing or droplet-based approaches[12].

To address the lack of dedicated scATAC-seq tool that enable simultaneous data integration (for example, batch correction) and dimensionality reduction, we have developed BAVARIA, a batch-adversarial variational auto-encoder (VAE)[13], which facilitates dimensionality reduction and integration for scATAC-seq data. To this end, we extended the standard VAE framework in several ways. First, inspired by combining deep learning with specialized and suitable count noise models for processing single-cell RNA-seq data (for example, by using a zero-inflated negative binomial distribution[14,15]), we set out to find a suitable count model for scATAC-seq. We identified the negative multinomial-based reconstruction loss to outperform alternative reconstruction losses with respect to extracting useful information about the underlying cell types, including the binary cross-entropy or the multinomial-derived loss (Extended Data Fig. 1 and Methods). The multinomial part of the reconstruction loss describes the accessibility profile across all regions as a whole, rather than considering the regions independently (for example as is assumed for the binary cross-entropy loss). This reduces the risk of obtaining a poor local minimum (for example, due to overfitting) and achieves invariance with respect to the read depth. The dispersion parameter, on the other hand, offers robustness against outliers during training. Second, fitting neural networks is commonly based on stochastic optimization, which may lead to variable latent feature quality across multiple training runs. Due to the optimizer getting stuck in a poor local optimum, cell types may occasionally be poorly characterized. Here, an ensemble approach, whereby latent features of several independently trained models are concatenated, not only stabilizes the latent feature quality, but also appears to improve their feature quality compared to latent features from individual models (Extended Data Fig. 2 and Methods). Third, we adopted a domain-adversarial training strategy[16] that encourages the VAE to extract latent features uninformative of batch effects. Specifically, we use batch-discriminator networks not only at the final layer of the encoder as suggested in Ganin et al.[16], but also at intermediate layers of the encoder (Fig. 1). This puts more emphasis on removing irrelevant batch-associated information in the initial layers of the network, which we find to enhance the batch correction capabilities (see comparison below and Methods). We refer to our approach as batch-adversarial VAE or BAVARIA (Fig. 1).

BAVARIA improves cell-type characterization

We systematically assessed the ability of BAVARIA-derived low-dimensional feature representations for cell-type characterization on a range of real and synthetic datasets. To this end,

[1]Max-Delbrück-Center for Molecular Medicine in the Helmholtz Association, Berlin Institute for Medical Systems Biology, Berlin, Germany. [2]Department of Biology, Humboldt University, Berlin, Germany. [3]Present address: Roche Diagnostics GmbH, Penzberg, Germany. ✉e-mail: wolfgang.kopp@mdc-berlin.de; uwe.ohler@mdc-berlin.de

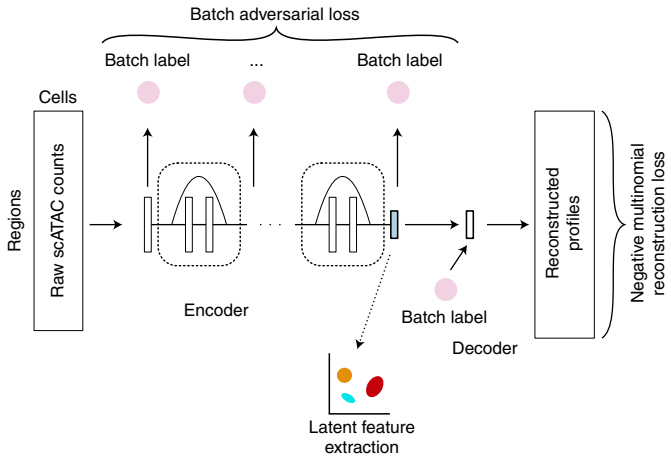


Fig. 1 | Schematic representation of BAVARIA. A variational auto-encoder utilizing a negative multinomial reconstruction loss and a batch-adversarial training strategy for data integration (BAVARIA). Latent features of the encoder network module serve as low-dimensional representation of the high-dimensional original accessibility profile.

we expanded on a recently published benchmarking framework[11] for comparision with current state-of-the-art solutions (cisTopic[4], LSI[5], SnapATAC[7], ArchR[6], scDEC[9], SCALE[8] and PeakVI[10]). Briefly, for each method, a low-dimensional feature representation is extracted and subjected to cell clustering. For BAVARIA, the low-dimensional representation is derived from the latent features of the encoder. Subsequently, a range of clustering evaluation scores are used to determine how well the cell clusters reflect cell identities based on known ground truth cell labels (if available) or by assessing the separation of known marker genes (Methods and ref. [11]).

Across three publicly available datasets, we observe variable performance of LSI, cisTopic, SnapATAC, SCALE, ArchR and PeakVI. Compared to these methods, scDEC seems to perform less well on the datasets. No single method consistently outperformed the other methods across the datasets (Fig. 2c,d). On the other hand, we find that BAVARIA robustly achieves best or competitive performance for uncovering cell identities across all real datasets (Fig. 2c,d). While PeakVI achieves a slightly better performance compared to BAVARIA on the Cusanovich 2018 subset when evaluated with the adjusted Rand Index (ARI) score, we find BAVARIA to perform similarly well or slightly better on that dataset for the adjusted mutual information (AMI) and homogeneity score (Hom). The performance of BAVARIA is also exemplified by the separation of known cell types in the uniform manifold approximation and projection (UMAP) embedding (Fig. 2b) and the high similarity of network-imputed and original signal tracks within distinct known cell types (Fig. 2a).

Next, we assessed the performance of the methods at different read depths and noise levels using two synthetic datasets (a bone marrow and an erythropoiesis dataset). We generated larger simulated datasets relative to ref. [11], as we found previous sizes to be insufficiently small to reflect current realistic experiments and for fitting large neural networks (Methods). For the synthetic bone marrow datasets, LSI and BAVARIA both achieve similar high performance across all sparsity levels, while the performance of cisTopic, SnapATAC, SCALE, PeakVI and scDEC decrease sooner with decreasing read depth (see 250 and 500 fragments per cell; Extended Data Fig. 3a). All methods perform similarly well on higher noise levels for the bone marrow dataset, except for scDEC which exhibited a systematically lower performance than the other tools and PeakVI which exhibited a moderate performance decrease for 40% noise (Extended Data Fig. 3b). On a synthetic erythropoiesis dataset, all methods achieve similar results for the high read coverage regime (see '5,000 fragments per cell'; Extended Data Fig. 3c). Yet, with decreasing read coverage, BAVARIA outperforms the other methods (see 1,000, 500 and 250 fragments per cell; Extended Data Fig. 3c). In addition, LSI and BAVARIA perform best for 0% and 20% additional noise, and BAVARIA outperforms all other methods for 40% additional noise using synthetic erythropoiesis data (Extended Data Fig. 3d).

Overall, in comparison to other methods we find that BAVARIA is robust and capable of extracting meaningful latent feature representations across a range of datasets.

Batch-adversarial training facilitates data integration

Having established the superior performance on benchmark tasks, we turned to BAVARIA's signature feature of data integration. We analysed two adult mouse brain cell samples from different sources: 10X Genomics[17] and Cusanovich et al.[5] (Methods). To quantify the contribution of our new data integration strategy, we first disabled adversarial training with BAVARIA. Here, cells from the two datasets occupy non-overlapping territories in the cell embedding space, which underlines the severity of the batch effect (Fig. 3a). That is, cells largely cluster by batches. By contrast, with batch-adversarial training, BAVARIA achieves a markedly better alignment between cells from different batches, while also largely maintaining a separation between previously annotated cell types (Fig. 3b). Compared to the originally proposed adversarial strategy by Ganin et al.[16] of using a single batch-discriminator network at the final layer of the encoder, we observe an improved cell-mixing performance when batch discriminators are placed not only on the final encoder layer, but also on the hidden encoder layers (Extended Data Fig. 4). In addition, we observe considerable batch effects when using a conditional VAE variant of BAVARIA in which one-hot encoded batch labels are used as additional inputs for the encoder (Extended Data Fig. 4)—similar to that proposed for scVI[15].

We compared BAVARIA against several batch integration methods (scVI, trVAE, SAUCIE, Harmony, Seurat v3 CCA, Liger and PeakVI). With the exception of PeakVI, the other tools were not specifically designed for processing scATAC-seq data. This enabled us to assess whether a dedicated approach to the characteristics of single-cell open chromatin would surpass a naive strategy to use a tool tailored to a different modality. Indeed we find that BAVARIA and PeakVI appear to achieve a reasonable separation between cell clusters, compared to tools that were not specifically designed for processing scATAC-seq data (Fig. 3b). For instance, with SAUCIE and trVAE different cell-types largely remain connected in the embedding. Harmony appears to have merged some cell types (see the centre-top in the Harmony panel of Fig. 3b). The integration with Liger has lead to a substantial number misalignments of 10X-derived cells with Cerebellar granule cells and a considerable amount of batch effects is still visible after the integration with Seurat (Fig. 3a,b). Not only does BAVARIA separate cell types reasonably well, but our model also yields a markedly better mixing of cells between batches compared to all other tools as measured by the kBET score and as is evident from the UMAP embeddings (Fig. 3a). In particular, we find that batch-conditional VAE models (for example scVI, PeakVI and the conditional variant of BAVARIA), are prone to leaving residual batch effects after the integration (Fig. 3a and Extended Data Fig. 4).

Next, we clustered the cells based on the joint latent features and inspected pseudo-bulk accessibility profiles stratified by the batches around several marker genes. For clusters with relatively even representation of cells from both batches, highly concordant accessibility profiles across clusters can be observed (for example, Scl17a7, Caln1, Gad2, Tmem119, Aldh1l1, Mbp and Abca4; Extended Data

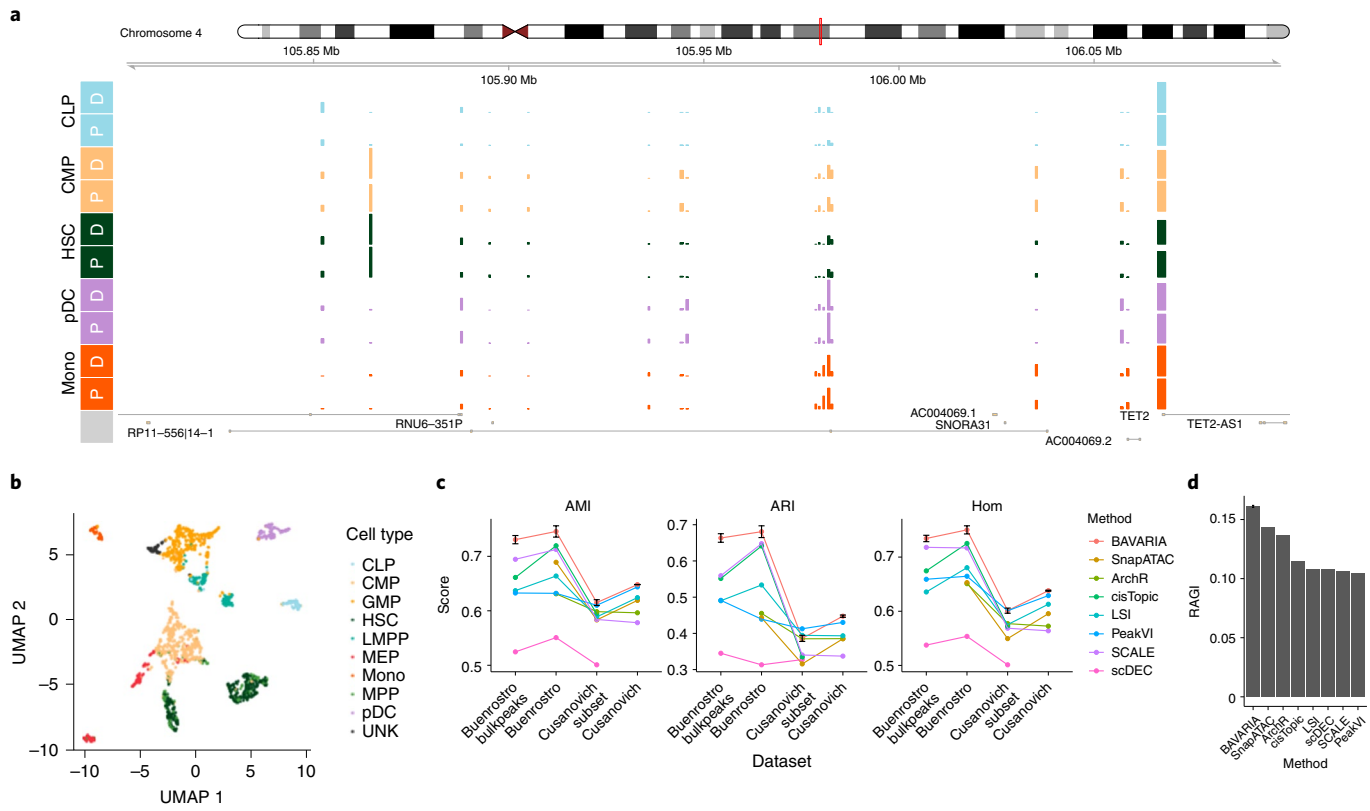


Fig. 2 | Comparison of dimensionality reduction methods. a, Comparison between observed and predicted pseudo-bulk accessibility profiles for the haematopoiesis data[19]. BAVARIA was fitted ten times from scratch. Predicted single-cell accessibility tracks were averaged across the models (excluding outlier models) and subsequently collapsed within the known cell types: CLP, CMP, HSC, pDC and mono cells. Pseudo-bulk tracks for the original input data and the prediction are indicated using the suffix D and P, respectively. **b**, Scatterplot of single cells from haematopoiesis data[19] in the UMAP space. Latent features were obtained by fitting BAVARIA ten times from scratch to mitigate fluctuations due to random initialization and concatenating the latent features of the individual models (excluding outlier models; Methods). The concatenated latent features were used as input to the UMAP algorithm. **c**, Cell clustering based on the derived low-dimensional feature representations were evaluated by comparing clusters against known ground truth cell labels on hematopoiesis data (Buenrostro 2018 and Buenrostro 2018 bulkpeaks[19]) and mouse tissue cells (Cusanovich 2018 and Cusanovich 2018 subset[5]) using the AMI, ARI and Hom. Generally, high scores indicate good behaviour for capturing known cell populations. Buenrostro 2018 and Buenrostro 2018 bulkpeaks use the same cells but different peak regions[11,19] and the Cusanovich 2018 subset consists of a subset of cells from Cusanovich 2018[11] (Methods). Results for LSI, cisTopic and SnapATAC were obtained from the benchmark assessment[11]. An ensemble of BAVARIA was run three times. The performances are summarized by the mean score (dot) and error bars that denote the ±s.e.m. **d**, Clustering performance evaluated on 10X Genomics 5k PBMCs[24] using the residual average Gini index (RAGI). which is based on marker gene separation[11]. Results for LSI, cisTopic and SnapATAC were obtained from the benchmark assessment[11]. An ensemble of BAVARIA was run three times. The performances are summarized by the mean score (bar) and error bars that denote the ±s.e.m.

Fig. 5a,b,d,e). This suggests that the latent features derived from BAVARIA are suitable for cell label transfer, as cells appear to cluster together based on their underlying cell type. On the other hand, as a sign for successful integration of data taken from similar but not identical sources, we also find cell clusters that are primarily present in one of the samples. For instance, cluster 0 and 18 consist of mostly Cusanovich 2018 cells which exhibit accessibility at Mmp24 and Itga11, whereas cluster 16 consists of mostly 10X Genomics cells which exhibit cluster-specific accessibility around Sh3bp4. While label transfer from an annotated reference onto a dataset without cell-type annotation is possible in a supervised manner (for example, by classifying cell types based on their accessibility profile), unsupervised data integration (for example, via BAVARIA) offers the possibility to correct and account for imperfections of the original cell annotations (for example, reference dataset). For instance, we observe several sub-populations of cells that have previously been annotated as inhibitory neurons (for example, clusters 4 and 14; Extended Data Fig. 5a). We do not find a separation of these cell clusters when using the other integration methods. Similarly, a cell population previously annotated as unknown (cluster 22; Extended Data Fig. 5c) likely represents doublet events between cerebellar granular cells and oligodendrocytes, as these cells exhibit specific accessibility for both cell types (compare clusters 0, 2 and 22; Extended Data Fig. 5c).

Computational requirements

We compared VAE-based models (SCALE, scVI, PeakVI and BAVARIA) and cisTopic in terms of runtime and memory requirements on a synthetic bone marrow dataset consisting of 12,000 cells and 80,000 peaks and a coverage of 5,000 reads per cell (based on the synthetic data from the benchmarking framework). The comparison was performed on a Linux server using an Intel Xeon Platinum 8168 CPU @ 2.70GHz processor with 3 TB RAM and an NVIDIA Tesla P40 GPU. The VAE-based models utilized a GPU for model fitting. All models were fitted for 100 epochs with otherwise default settings. For BAVARIA, a single VAE was fitted. A cisTopic model was fitted with 10 topics using CPUs.

The runtime for cisTopic is markedly higher than the deep learning-based methods, as the latter set of methods benefit from GPU-accelerated processing (Supplementary Table 1).

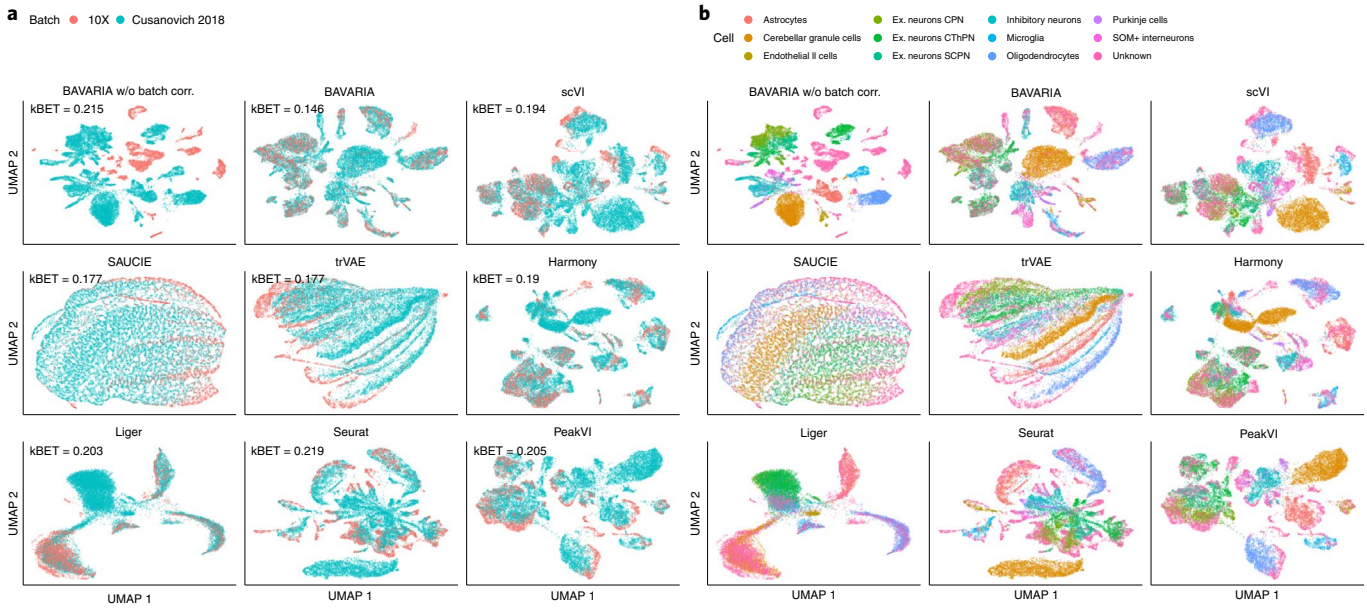


Fig. 3 | Comparison of data integration methods. a, UMAP embedding illustrating cells from 10X Genomics and Cusanovich 2018 after applying BAVARIA without batch correction, BAVARIA, scVI[15], SAUCIE[26], trVAE[25], Harmony[27], Seurat[30], Liger[29] and PeakVI[10]. The kBET scores indicate the cell mixing across batches[31]. Lower kBET scores correspond to better mixing. **b**, UMAP embedding illustrating previously characterized cell types[5] (astrocytes, cerebellar granule cells, encothelial II cells, Excitatory (Ex.) neurons CPN, Ex. neurons CThPN, Ex. neurons SCPN, inhibitory neurons, microglia, oligodendrocytes, Purkinje cells, SOM+ interneurons and unknown cells). 10X Genomics cells[17] are labelled 'unknown' as no labels are available.

Fitting a single VAE within BAVARIA requires similar or slightly lower runtime compared to the other VAE-based methods (Supplementary Table 1).

All methods handle data in the form of sparse matrices, which is critical for scATAC-seq processing and in general leads to similar memory footprints. For BAVARIA, we make use of tensorflow data pipelines[18] to optimize mini-batch processing during training and evaluation. This, however, introduces a higher memory overhead compared with the other methods (Supplementary Table 1).

In general, the memory and runtime requirements of BAVARIA depend linearly on the number of cells and the number of peaks. Additionally, the runtime depends linearly on the number of training epochs and the ensemble size. For large datasets, a trade-off between speed and accuracy can be achieved by (1) adjusting the ensemble size, (2) by subsetting the cells for model fitting and/or (3) by subsetting the available features (for example, peaks).

Conclusion

In summary, we have developed a VAE for integrating sparse and high-dimensional scATAC-seq data (BAVARIA). We demonstrated that several unique aspects, regarding robust training and noise modelling for scATAC count data, allow the model to match and exceed the performance of current state-of-the-art solutions for cell type characterization across (1) different dataset sizes, (2) different read depths and (3) different noise levels. Importantly, its batch-adversarial training strategy makes BAVARIA the first tool to facilitate data integration and accurate batch correction across different scATAC protocols.

Methods

Benchmark analysis. Data for the benchmark analysis was obtained by following and adapting a recently published scATAC-seq benchmarking framework[11]. We obtained (1) a haematopoietic differentiation dataset ($n = 2,034$ cells; Buenrostro 2018[19]), (2) a single-nucleus combinatorial indexing ATAC-seq mouse tissue dataset ($n = 81,173$ cells; Cusanovich 2018[5]) and (3) a peripheral blood mononuclear cell dataset ($n = 5,335$ cells; 10X peripheral blood mononuclear cells (PBMC) 5k). The former two datasets also include ground truth cell labels from fluorescence-activated cell sorting (FACS) sorted cells or by using the tissue of origin as label. The benchmark also includes a 15% subset of cells from the mouse tissue dataset ($n = 12,178$ cells; Cusanovich 2018 subset[5]).

The haematopoietic dataset was independently processed twice with two different sets of peaks. Namely, peaks determined on the scATAC-seq by Chen et al.[11] (Buenrostro 2018) and the original peak set reported by Buenrostro et al.[19] (Buenrostro 2018 bulkpeaks).

Preprocessing includes feature counting in the pre-defined peak regions, binarization of the count matrix, filtering for a minimum read coverage per peak (at least 1% of cells need to be covered at the region), and removing sex chromosomes. This led to 98,738, 132,110, 378,894, 141,388 and 67,427 peaks for the Buenrostro 2018, Buenrostro 2018 bulkpeaks, Cusanovich 2018, Cusanovich 2018 subset and 10X datasets, respectively.

We followed the procedure of ref. [11] to generate several synthetic datasets based on FACS-sorted bulk-ATAC-seq samples from bone marrow[20] and erythropoiesis [21], as described previously[11]. As the originally published benchmarking assessment consists of too few cells for fitting large neural networks, we increased the numbers of cells. Specifically, for bone marrow, 2,000 cells per population were generated with different fragment sizes per cell (5,000, 2,500, 1,000, 500 and 250 fragments) as well as for different noise levels (0%, 20% and 40% additional noisy reads). Likewise, for erythropoiesis, 1,000 cells per population were generated with different fragment sizes per cell (5,000, 2,500, 1,000, 500 and 250 fragments) as well as for different noise levels (0%, 20% and 40% noise). Note also that the downsampling experiment based on the erythropoiesis data was not part of the original benchmarking assessment[11]. Finally, for all synthetic datasets, the 80,000 most covered regions were retained for the benchmark analysis.

Mouse brain cell integration. We downloaded fresh adult mouse brain cell data from the 10X Genomics web site (https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_adult_brain_fresh_5k) as well as scATAC-seq data from several mouse tissues[5] from https://atlas.gs.washington.edu/mouse-atac/data/.

We used the peaks provided by 10X Genomics as reference peaks[17]. The master peaks from Cusanovich et al.[5] were lifted over from mm9 to mm10 and mapped onto the reference peaks using bedtools intersect[22]. Likewise, the original count matrix was mapped onto the new reference peak set and only cells from the WholeBrain and PreFrontalCortex tissues were retained for the analysis[5]. The count matrices from the 10X Genomics and Cusanovich 2018 datasets were concatenated, binarized and filtered to ensure that each region was covered in at least 1% of the cells. Regions on the sex chromosomes were removed. The final count matrix contained 18,605 cells (3,880 from 10X and 14,725 from Cusanovich et al.[5]) and a peak set of size 136,528.

Negative multinomial variational auto-encoder. We define the accessibility profile of a given single cell across a set of regions as $x=[x_1,...,x_N]$ where $x_i>0$ reflects accessibility of region i. $x_i=0$ indicates inaccessibility. In the context of this work, we use binarized accessibility profiles (for example, $x_i=0$ or $x_i=1$), although the model is in general also applicable to non-negative integer vectors (for example, sites with multiple reads per cell).

In this section, we first introduce the adaptation of the standard VAE model[13] without integrated batch-correction and turn to batch-adversarial training strategy in the next section. Briefly, following Kingma and Welling[13], the encoder determines the L-dimensional mean μ and variance σ parameters of the approximate Gaussian posterior distribution of the latent feature representation given an accessibility profile x. From the approximate posterior distribution, samples, z, are drawn, which are in turn used as input for the decoder to reconstruct the N dimensional accessibility profile[13]. We shall use the mean vector, μ, as the low-dimensional feature representation used for the downstream analysis.

We assume that the accessibility profile, x, follows a negative multinomial distribution defined by

$$P(x) := \Gamma\left(r+\sum_{i=1}^{N} x_i\right)\frac{p_0^r}{\Gamma(r)}\prod_{i=1}^{N}\frac{p_i^{x_i}}{x_i!}$$

where $p=[p_0,\ldots,p_N]$ represent the non-negative parameters which sum to one and r denotes the positive real-valued dispersion parameter. p_i for $i>0$ reflects the accessibility profile, while p_0 is associated with the dispersion.

We construct a decoder that determines the respective parameters p and r. In particular, for $i>0$ the decoder computes

$$p_i = \frac{\exp(a_i(z))}{1+\sum_{j=1}^{N}\exp(a_j(z))}$$

where a_i represents the decoder's output activity at region i for a given latent feature sample z. The remaining probability mass is reserved for the dispersion and is given by

$$p_0 = \frac{1}{1+\sum_{j=1}^{N}\exp(a_j(z))}.$$

Here we assume that the dispersion parameter is scalar and does not depend on the accessibility profile x, that is, it is adjusted like a bias term in the network.

Consequently, the reconstruction loss is given by

$$\text{loss}_{\text{recon}} := -\log\Gamma(r+\sum_{i=1}^{N}x_i) - r\log p_0(z) + \log\Gamma(r) - \sum_{i=1}^{N}x_i\log p_i(z).$$

As described previously[13], Kullback–Leibler divergence is utilized as a regularization loss, denoted as loss_{KL}, which encourages that the latent feature representation is distributed according to $\mathcal{N}(0, I)$. The total loss is given by summing the reconstruction loss and the regularization loss, which is subject to minimization by adapting the model parameters during the model fitting.

Batch-adversarial training. To facilitate data integration and batch correction, we took inspiration from ref. [16] and adapted the variational auto-encoder framework described above to enable batch-adversarial training. The goal of this approach is to establish a latent feature representation that captures biologically relevant information about the accessibility profiles (for example, to describe cell types), while at the same time conveying as little information as possible about experimental batch labels.

To facilitate batch-adversarial training, the standard VAE architecture is augmented by batch-discriminator network modules. These sub-networks are stacked on top of the final layer of the encoder for the purpose of predicting the batch label from the latent feature representation. We use batch-discriminator networks with softmax output activation to predict categorical batch labels. Multiple independent batch labels can be used simultaneously with separate softmax output units for each batch. In addition to the batch-discriminator network at the final layer of the encoder (similar to ref. [16]), we also add batch discriminators on top of the hidden layers of the encoder (after the first hidden layer and after each residual network block). The batch-discriminator network modules at the initial and intermediate layer allow to put more emphasis on removing batch related information early on in the network, and is intended to simplify disentangling batch-related information from the biologically relevant signal throughout the entire encoder network.

Apart from the additional batch-discriminator networks, we modified the decoder to take as input the latent feature encoding z as well as the one-hot encoded batch labels. This enables the decoder to recombine the batch information with the (ideally) batch-corrected latent feature representation to compute the reconstruction of the accessibility profiles. This is important, as batch-related information is still used in this way to compute the reconstruction loss.

Finally, we adjust the training objective of the standard VAE as follows: We measure how well batch-labels can be predicted from the latent features of the encoder (for example, derived from the hidden or final layer) using the categorical cross-entropy loss

$$\text{loss}_{\text{batch}} := -\sum_i y_i\log\hat{y}_i$$

where y and $\hat{y}$ denote the true and predicted batch label. While the parameters of the batch-discriminator associated parameters are adapted to minimize $\text{loss}_{\text{batch}}$, the parameters associated with the encoder module are adapted according to $\text{loss}_{\text{BAVARIA}}=\text{loss}_{\text{recon}}+\text{loss}_{\text{KL}}-\text{loss}_{\text{batch}}$ That is, the encoder seeks to find a latent feature representation that is uninformative for the batch-label classification.

Model and training hyper-parameters. We use the following model architecture for all experiments: for the encoder, we use a feed-forward layer with 512 nodes and rectified linear units (ReLU) activation, followed by 20 consecutive residual neural network blocks with 512 nodes, which feed into two layers representing the means and variances of the latent features (dimensions listed in Supplementary Table 2). Each residual block is composed of a feed-forward layer with ReLU and a feed-forward layer whose output is added to the block's input before applying ReLU activation. For the decoder, we use a single feed-forward layer consisting of 16 and 25 neurons in the benchmarking analysis and the data integration use case, respectively.

For the batch-adversarial training, batch discriminator network modules are stacked on top of the intermediate layers of the encoder (after the first layer and after each residual block) as well as on top of the final layer of the encoder. Each discriminator network consists of two layers with 128 neurons and ReLU activation and an output layer with softmax activation.

The models were fitted using 85% of the cells using AMSgrad[23]. The remaining 15% cells were used for validation. Additional dataset-specific hyper-parameters are listed in Supplementary Table 2.

Ensemble of models and feature extraction. We fitted a BAVARIA model M times, each time starting from random initial weights. Afterwards, we concatenated the mean vectors of the approximate posterior distribution of the latent features μ either across all M individual models or by using a subset of these models, dependent on the use case. In the latter case, we sought to remove potentially poor quality models whose average loss across the dataset exceeded an outlier criterion. Specifically, we removed models if their average loss after training exceeded $Q_{75\%}(\text{loss})+1.5\times\text{IQR}(\text{loss})$, where $Q_{75\%}$ denotes the 75% quantile of the loss distribution across the M individual models and IQR represents its interquartile range. The latent features of the remaining models were concatenated and considered for the downstream cell clustering analysis.

Benchmark analysis. We adapted a recently published scATAC-seq benchmarking framework[11]. For all real datasets, we obtained the results of three top-performing tools: cisTopic[4], LSI[5] and SnapATAC[7], as previously reported[11]. The results of the remaining tools from the original benchmarking assessment are omitted here. In addition, we ran LSI on the full Cusanovich 2018 dataset in the same way it was previously run for the Cusanovich 2018 subset[11]. We fitted BAVARIA models on the filtered count matrices using the hyper-parameters described in Supplementary Table 2. After training each individual model, an ensemble model was created by concatenating individual models by excluding outlier models. Latent features of the ensemble were used for the downstream clustering using the benchmarking framework. We ran SCALE[8] with default parameters using the input count matrices described above (for example, the same matrices which were used for BAVARIA) and extracted the latent features for the downstream clustering analysis. We created arrow files using ArchR[6] for each dataset based on the bam-files (merged across cells) that are part of the benchmarking framework without additional filtering. For each dataset iterative LSI was performed as demonstrated in the online tutorial https://www.archrproject.com/articles/Articles/tutorial.html. Subsequently, the reduced dimensionality was evaluated using the benchmarking framework[11]. We applied PeakVI[10] on each dataset (with the same filtering criteria as were used for applying BAVARIA) by following the online tutorial https://docs.scvi-tools.org/en/stable/user_guide/notebooks/PeakVI.html. Similarly, we applied scDEC[9] by following the online documentation https://github.com/kimmo1019/scDEC. For the parameter –K we used 8, 10, 10, 13 and 13 clusters for the 10X PBMC, Buenrostro 2018, Buenrostro 2018 (bulkpeaks), Cusanovich 2018 (subset) and Cusanovich 2018 (full), respectively. Moreover, we used 15 latent features for all cases, except for Cusanovich 2018 full where we used 30 latent features.

For the synthetic datasets, we ran cisTopic, LSI and SnapATAC with the same parameters as in the published benchmark analysis, but using larger numbers of cells. We applied SCALE with parameters '–min_cells 0 –min_peaks 0.0' to ensure that the simulated count matrix is not further subjected to filtering. We applied PeakVI as described above. We used scDEC with 15 latent features and 8 clusters as described above. BAVARIA ensembles were fitted for each synthetic dataset using hyper-parameters listed in Supplementary Table 2.

Following ref. [11], the latent features extracted for each method were subjected to k-means clustering, hierarchical clustering (with ward linkage), and Louvain clustering. For all datasets with available ground truth labels, the ARI, AMI and

Hom were used to score the agreement of the clustering results with the known cell labels[11]. For the 10X dataset (without ground truth)[24], the residual average Gini index (RAGI) was computed based on a previously reported set of marker and house-keeping genes[11].

Subsequently, the maximum clustering score across the clustering algorithms (k-means clustering, hierarchical clustering or Louvain clustering) was reported for each dataset and score (ARI, AMI, Hom).

Comparison of alternative reconstruction losses. We compared several reconstruction loss implementations on Buenrostro 2018 data using the benchmarking framework described above[11]. Specifically, we used the same variational auto-encoder architecture parameters, with exception of the final layer of the decoder. Assuming the accessibility profiles are derived from a Bernoulli distribution, we used a sigmoid output activation for the decoder in conjunction with a binary cross-entropy loss, $\text{loss}_{\text{Bin}} = -\sum_i (x_i \log(p_i(x)) + (1 - x_i)\log(1 - p_i(x)))$. Alternatively, assuming multinomially distributed accessibility profiles, we used a softmax output activation in conjunction with a negative log-likelihood of the multinomial distribution, $\text{loss}_{\text{Mul}} = -\sum_i x_i \log p_i(x)$.

Data integration of mouse brain cells. We used BAVARIA with a 15-dimensional latent feature representation and 25 neurons for the hidden layer of the decoder. We trained an ensemble of 10 individual models for 200 epochs using a batch size of 64 (Supplementary Table 2). In addition, each individual model was fitted on a random 50% subset of the original peak set, which enabled training time speedup while maintaining similar clustering qualities (data not shown).

We compared BAVARIA to several tools that facilitate batch correction. We fitted scVI[15] with default parameters. We trained trVAE[25] by following the tutorial code (https://nbviewer.jupyter.org/github/theislab/trVAE/blob/master/examples/trVAE_Haber.ipynb) using the unnormalized count matrix (136,528 regions by 18,605 cells). We fitted SAUCIE[26] with default parameters. We fitted PeakVI[10] with default parameters. We used read depth normalized (with a target of 10,000 reads per cell) and $\log(x + 1)$ transformed signals, determined 50 principal components and integrated the datasets with Harmony[27] using the scanpy interface[28]. Integration with Liger[29] was performed by following the tutorial code https://github.com/welch-lab/liger/blob/master/vignettes/Integrating_multi_scRNA_data.html. Integration with Seurat v3 based on canonical correlation analysis (CCA)[30] was performed by following the tutorial code https://satijalab.org/seurat/articles/integration_introduction.html using all available features.

As baseline, we ran simplified versions of the BAVARIA architecture: (1) we adjusted BAVARIA to a conditional VAE model, which receives the batch labels as input at the encoder (along with the raw read profile), rather than predicting the batch labels from the latent features and hidden layers; (2) we adjusted BAVARIA to use only a single batch-discriminator network module at the final layer of the encoder (along the lines of Ganin et al. 2016[16]). These networks were trained with the same network parameters and hyper-parameters as BAVARIA.

The UMAP embedding was computed based on the latent features from each method using scanpy[28]. For the UMAP visualization, cells previously annotated as astrocytes, cerebellar granule cells, encothelial II cells, Ex. neurons CPN, Ex. neurons CThPN, Ex. neurons SCPN, inhibitory neurons, microglia, oligodendrocytes, Purkinje cells, SOM+ interneurons and unknown cells in Cusanovich et al. 2018 and all 10X Genomics cells[17] were illustrated.

We determined kBET scores using the parameters k0 = 10 and n_repeat = 500 using the kBET R package[31] to measure the mixing of cells across batches.

Clustering and differential accessibility analysis of mouse brain data. Using the BAVARIA-derived latent features, we performed Louvain clustering using scanpy[28]. Cluster-specific accessibility was determined by using a generalized linear model and a binomial distribution:

$$\log p_{rcb} = \alpha_r + \beta_{cb} + \delta_{rc} + f_{rb}$$

where α_r denotes the region-specific offset for region r, β_{cb} denotes the cluster and batch-specific offset for cluster c and batch b, δ_{rc} denotes the cluster-specific accessibility for region r and cluster c and f_{rb} denotes the batch-specific accessibility for region r and batch b.

After fitting the linear models, we identified the 100 top-most accessible regions per cluster by ranking δ_{rc} and visualized the associated raw accessibility profiles in a heatmap using scanpy[28].

For the pseudo-bulk visualization, we re-mapped reads from Cusanovich et al.[5] WholeBrain and PreFrontalCortex tissues to mm10 using bowtie2[32] using the parameters '–very-sensitive –X 2000 –3 1'. Cluster-specific pseudobulk bam files were constructed by dividing the reads by barcodes associated with the clusters. These bam files were converted to bigwig tracks using 'bamCoverage –normalizeUsing CPM' from deeptools[33]. Finally, pyGenomeTracks[34] was used to visualize the cluster-specific accessibility tracks.

Data availability

All datasets to perform the benchmark analysis were obtained from the computational scATAC-benchmarking framework at https://github.com/pinellolab/scATAC-benchmarking. This includes the publicly available scATAC-seq 10X PBMC 5k dataset[24], the haematopoiesis dataset[19] and the adult mouse dataset[5], as well as the bone marrow[20] and erythropoiesis datasets[21] from which the simulated scATAC-seq datasets were derived. For the brain data integration, we additionally obtained mouse brain cells from Cusanovich et al.[5] https://atlas.gs.washington.edu/mouse-atac/data/ and the 10X Genomics website https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_adult_brain_fresh_5k.

Code availability

BAVARIA is available via GitHub under a GPL-v3 licence at https://github.com/BIMSBbioinfo/bavaria[35].

Received: 19 May 2021; Accepted: 11 January 2022;
Published online: 23 February 2022

References

1. Zamanighomi, M. et al. Unsupervised clustering and epigenetic classification of single cells. *Nat. Commun.* **9**, 2410 (2018).
2. Schep, A. N., Wu, B., Buenrostro, J. D. & Greenleaf, W. J. chromvar: inferring transcription-factor-associated accessibility from single-cell epigenomic data. *Nat. Methods* **14**, 975–978 (2017).
3. Baker, S. M., Rogerson, C., Hayes, A., Sharrocks, A. D. & Rattray, M. Classifying cells with scasat, a single-cell ATAC-seq analysis tool. *Nucleic Acids Res.* **47**, 10 (2019).
4. González-Blas, C. B. et al. cistopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nat. Methods* **16**, 397–400 (2019).
5. Cusanovich, D. A. et al. A single-cell atlas of in vivo mammalian chromatin accessibility. *Cell* **174**, 1309–1324 (2018).
6. Granja, J. M. et al. ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nat. Genet.* **53**, 403–411 (2021).
7. Fang, R. et al. Comprehensive analysis of single cell ATAC-seq data with SnapATAC. *Nat. Commun.* **12**, 1337 (2021).
8. Xiong, L. et al. SCALE method for single-cell ATAC-seq analysis via latent feature extraction. *Nat. Commun.* **10**, 4576 (2019).
9. Liu, Q., Chen, S., Jiang, R. & Wong, W. H. Simultaneous deep generative modelling and clustering of single-cell genomic data. *Nat. Mach. Intell.* **3**, 536–544 (2021).
10. Ashuach, T., Reidenbach, D. A., Gayoso, A. & Yosef, N. PeakVI: a deep generative model for single cell chromatin accessibility analysis. Preprint at *bioRxiv* https://doi.org/10.1101/2021.04.29.442020 (2021).
11. Chen, H. et al. Assessment of computational methods for the analysis of single-cell ATAC-seq data. *Genome Biol.* **20**, 241 (2019).
12. Luecken, M. D. et al. Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).
13. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations* (2014).
14. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
15. 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).
16. Ganin, Y. et al. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**, 2096–2030 (2016).
17. Fresh cortex from adult mouse brain (P50). *10X Genomics* https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_adult_brain_fresh_5k (2019).
18. Abadi, M. et al. TensorFlow: large-scale machine learning on heterogeneous systems (TensorFlow, 2015); https://www.tensorflow.org/
19. Buenrostro, J. D. et al. Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. *Cell* **173**, 1535–1548 (2018).
20. Ulirsch, J. C. et al. Interrogation of human hematopoiesis at single-cell and single-variant resolution. *Nat. Genet.* **51**, 683–693 (2019).
21. Ludwig, L. S. et al. Transcriptional states and chromatin accessibility underlying human erythropoiesis. *Cell Rep.* **27**, 3228–3240 (2019).
22. Quinlan, A. R. & Hall, I. M. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010).
23. Reddi, S. J., Kale, S. & Kumar, S. On the convergence of Adam and beyond. In *International Conference on Learning Representations* (2018).
24. 5k peripheral blood mononuclear cells (PBMCs) from a healthy donor. *10X Genomics* https://support.10xgenomics.com/single-cell-atac/datasets/1.0.1/atac_v1_pbmc_5k (2018).
25. Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Wolf, F. A. Conditional out-of-distribution generation for unpaired data using transfer vae. *Bioinformatics* **36**, 610–617 (2020).
26. Amodio, M. et al. Exploring single-cell data with deep multitasking neural networks. *Nat. Methods* **16**, 1139–1145 (2019).

27. Korsunsky, I. et al. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. Methods* **16**, 1289–1296 (2019).
28. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 1–5 (2018).
29. Welch, J. D. et al. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* **177**, 1873–1887 (2019).
30. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 (2019).
31. Büttner, M., Miao, Z., Wolf, F. A., Teichmann, S. A. & Theis, F. J. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods* **16**, 43–49 (2019).
32. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**, 357–359 (2012).
33. Ramírez, F., Dündar, F., Diehl, S., Grüning, B. A. & Manke, T. deepTools: a flexible platform for exploring deep-sequencing data. *Nucleic Acids Res.* **42**, 187–191 (2014).
34. Lopez-Delisle, L. et al. pyGenomeTracks: reproducible plots for multivariate genomic datasets. *Bioinformatics* **37**, 422–423 (2020).
35. Kopp, W., Akalin, A. & Ohler, U. BAVARIA source code v0.1. *Zenodo* https://doi.org/10.5281/zenodo.5791250 (2021).

Acknowledgements

We thank P. Rautenstrauch, J. Ronen and R. Monti for valuable comments on the manuscript. This work was supported by the German Federal Ministry of Education and Research (de.NBI; FKZ 031L0101B) and by the Helmholtz Association (sparse2big; ZT-I-0007).

Author contributions

W.K. designed and implemented BAVARIA and performed the analysis with input from U.O. A.A. provided resources dedicated to the project. W.K. and U.O. wrote the manuscript.

Funding

Open access funding provided by Max-Delbrück-Centrum für Molekulare Medizin in der Helmholtz-Gemeinschaft (MDC).

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-022-00443-1.

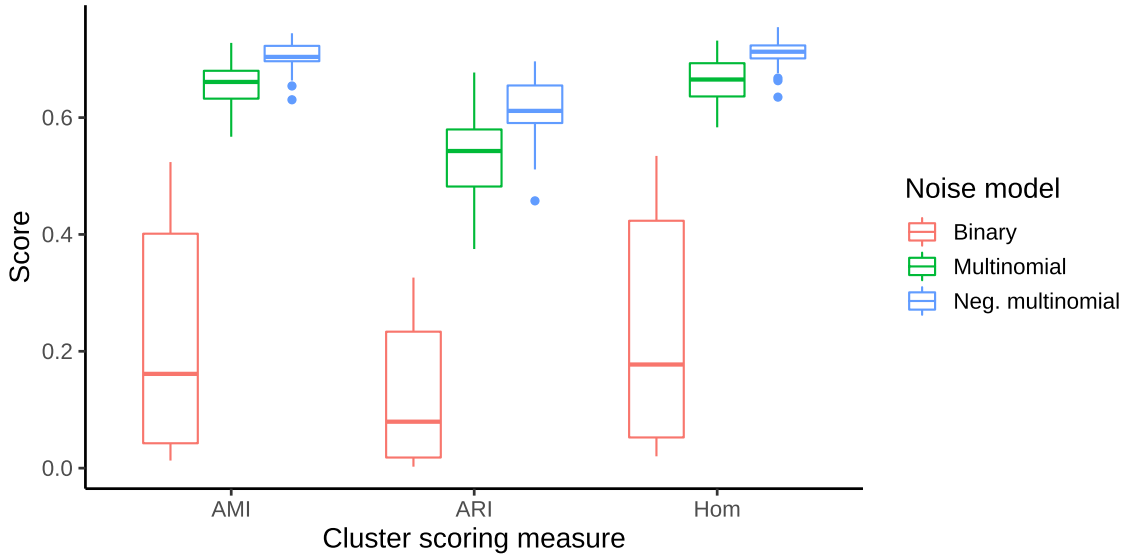
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-022-00443-1.

Correspondence and requests for materials should be addressed to Wolfgang Kopp or Uwe Ohler.

Peer review information *Nature Machine Intelligence* thanks Bo Li and Wing Wong for their contribution to the peer review of this work.

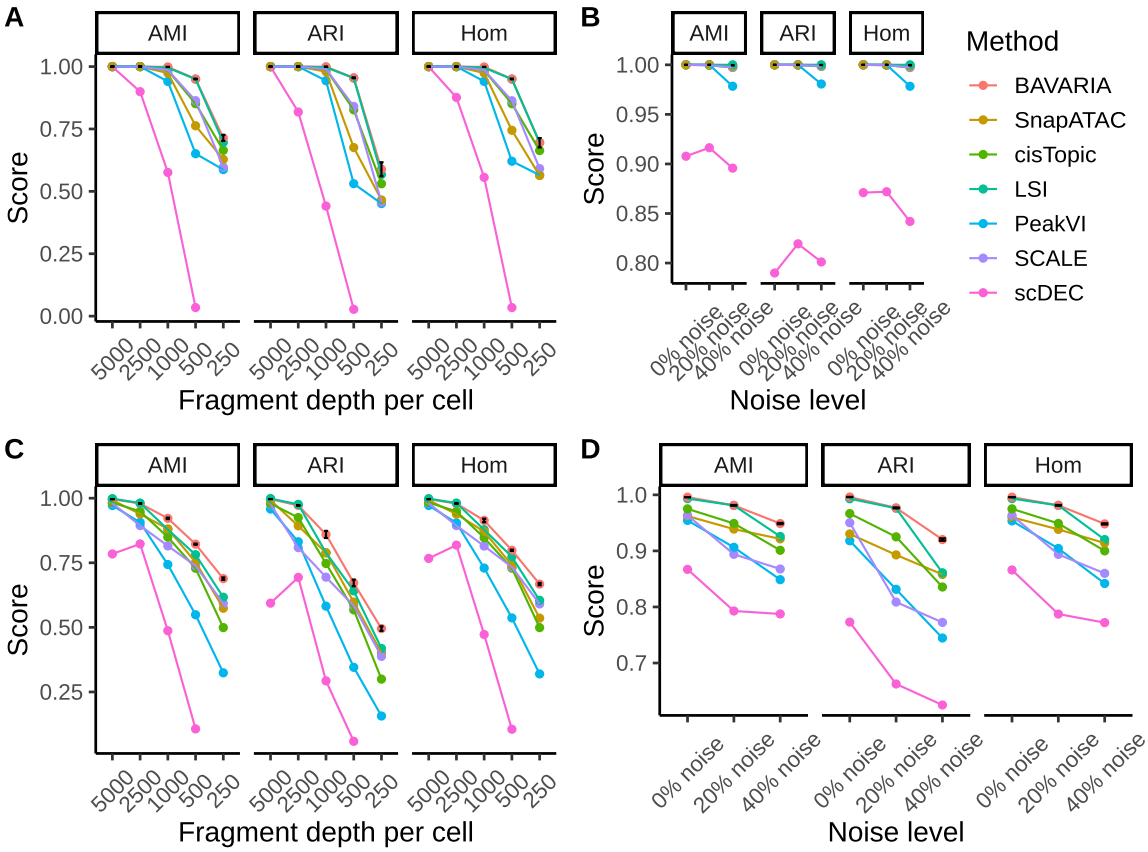
Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

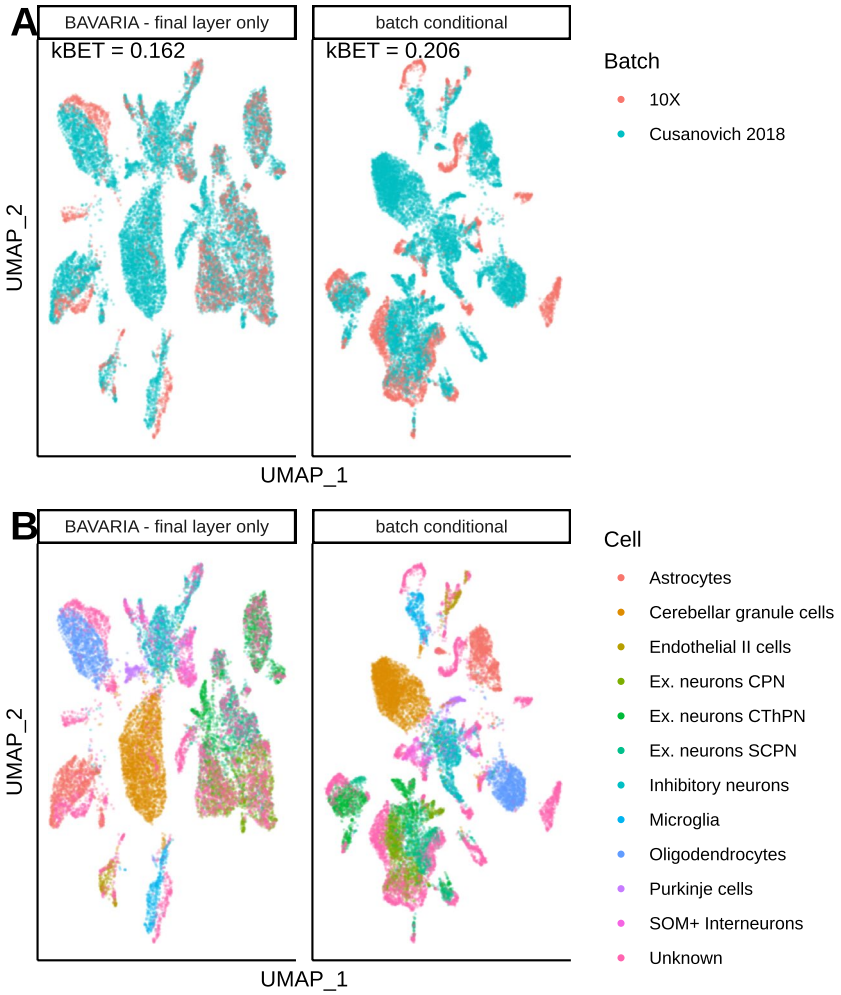


Extended Data Fig. 1 | Comparison of reconstruction loss measures. The suitability of different reconstruction loss measures was assessed by fitting thirty individual models on the Buenrostro et al. 2018 dataset. The total loss across the dataset was determined for each model and models with poor outlier losses were excluded (for example due to poor local minima; see Methods), leading to 28, 29 and 29 models for binary, multinomial and negative multinomial loss for the visualization, respectively. The x-axis represents different reconstruction losses: binary cross-entropy loss (Binary), negative log-likelihood for the multinomial distribution (Multinomial), and negative log-likelihood of the negative multinomial (Neg. multinomial) distribution. Otherwise, the model architecture remained the same. Latent features were subjected to clustering using k-means, hierarchical clustering and Louvain clustering and clustering performances were computed based on adjusted mutual information (AMI), adjusted Rand index (ARI) and Homogeneity (Hom) against ground truth cell labels. The best score across the clustering algorithms was considered. Boxes represent quartiles Q1 (25% quantile), Q2 (median) and Q3 (75% quantile); whiskers comprise data points that are within 1.5 x IQR (inter-quartile region) of the boxes.

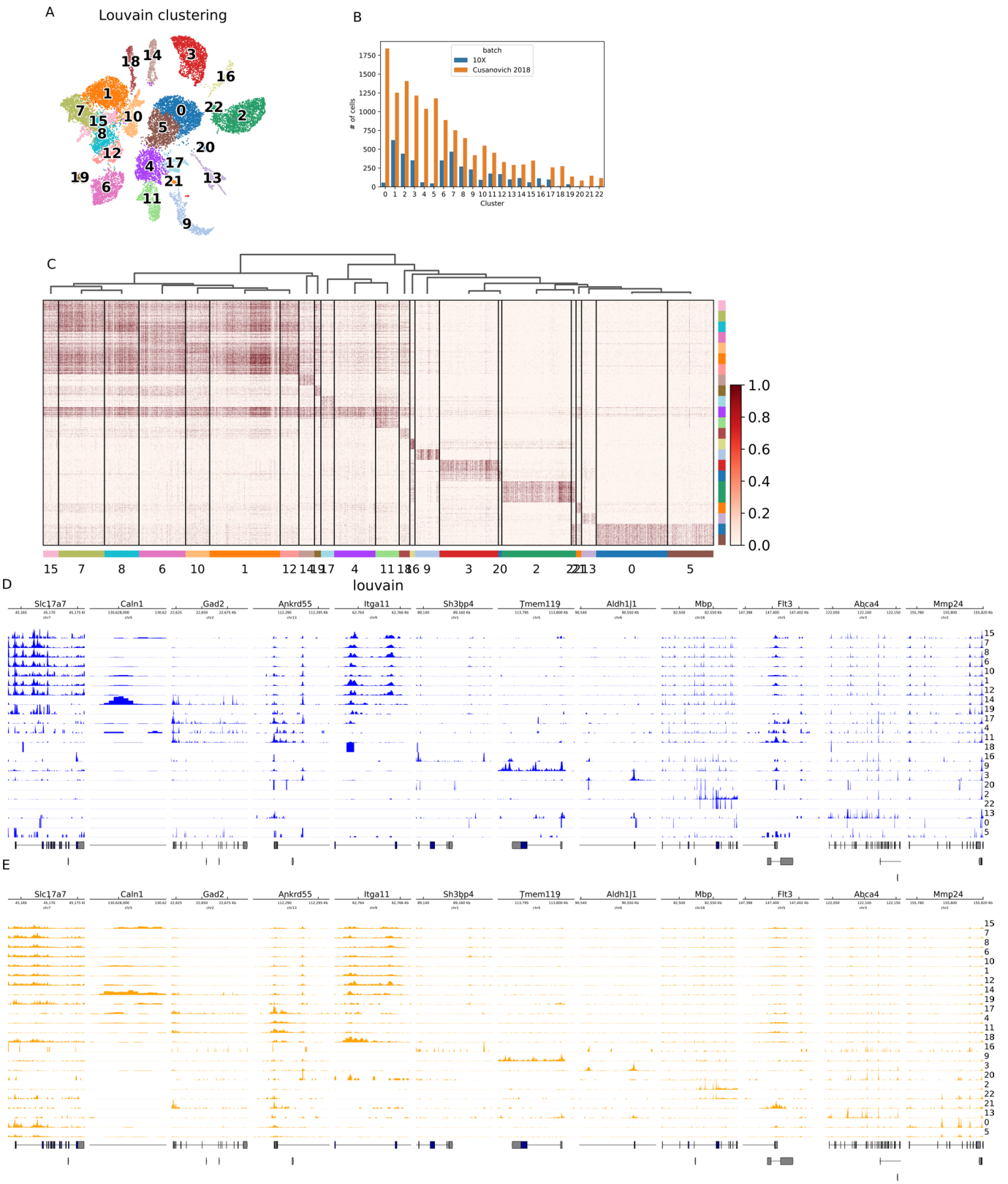
Extended Data Fig. 2 | Combining latent features of separately trained models. Three ensembles consisting of ten VAE models were fitted on the Buenrostro et al. 2018 dataset. The total loss across the dataset was determined for each model and models with poor outlier losses were excluded from the ensemble (for example, due to poor local minima; see Methods). Individual models (BAVARIA - individual) were combined to ensembles by concatenating the latent features (BAVARIA - ensemble). Latent features were subjected to clustering using several algorithms and clustering performances were computed based on adjusted mutual information (AMI), adjusted Rand index (ARI) and Homogeneity (Hom). The best score across the clustering algorithms are considered.



Extended Data Fig. 3 | Cell type characterization assessment using synthetic data. A) Bone marrow data using 5000, 2500 1000, 500 and 250 fragments per cell. B) Bone marrow data using 0%, 20% and 40% additional noise. C) Erythropoiesis data using 5000, 2500 1000, 500 and 250 fragments per cell. D) Erythropoiesis data using 0%, 20% and 40% additional noise. Low-dimensional feature representations were obtained using cisTopic, LSI, SnapATAC, SCALE and BAVARIA and subjected to clustering using different algorithms (k-means, hierarchical clustering, Louvain clustering). Clustering performances were evaluated using adjusted mutual information (AMI), adjusted Rand index (ARI) and Homogeneity (Hom) compared against ground truth cell labels (see Methods). The best score across clustering algorithms is shown. cisTopic, LSI, SnapATAC and SCALE were run once per case, while $N = 3$ ensembles of NM-VAE were trained from scratch to assess the variability of the performance. The dot represents the mean performance and the error bars indicate the +/- SEM according to the repetitions.



Extended Data Fig. 4 | Batch correction - Comparison of architectures. A) UMAP embedding illustrating cells from 10X Genomics and Cusanovich et al. 2018 after applying a BAVARIA variant with a single batch-discriminator network module at the final encoder layer (BAVARIA - final layer only) and a conditional variational auto-encoder variant of BAVARIA which receives the batch labels as input for the encoder's initial layer (batch conditional). The kBET scores indicate the cell mixing across batches. Lower kBET scores correspond to better mixing. B) UMAP embedding illustrating previously characterized cell types (Astrocytes, Cerebellar granule cells, Encothelial II cells, Ex. neurons CPN, Ex. neurons CThPN, Ex. neurons SCPN, Inhibitory neurons, Microglia, Oligodendrocytes, Purkinje cells, SOM+ interneurons and unknown cells; Cusanovich et al. 2018). 10X Genomics cells are labelled 'Unknown'.



Extended Data Fig. 5 | Clustering and cluster-associated regions. A) Clustering of the integrated 10x and Cusanovich et al. 2018 datasets. B) Number of cells per cluster and batch. C) Illustration of cluster-associated accessibility using the 100 top accessible regions per cluster. D) Depth normalized accessibility tracks per cluster for the 10X dataset for several marker regions. E) Depth normalized accessibility tracks per cluster for the Cusanovich et al. 2018 dataset for several marker regions.