UHH Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Applications of deep generative modeling approaches on Omics data

Dissertation zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Biologie
der Universität Hamburg
vorgelegt von Fabian Hausmann
Hamburg, Januar 2023

Vorsitzender der Prüfungskommission

Professor Dr. Tobias Lenz

Gutachter

Professor Dr. Stefan Kurtz
Professor Dr. Stefan Bonn

Datum der Disputation

07.07.2023

Abstract

Single-cell RNA-sequencing (scRNA-seq) provides detailed insights into the biology of tissue and disease development on the level of single cells. This cell-specific information can be used for cell identification, inference of cell development and disease characterization. However, current sequencing methods suffer from technical constraints, especially large differences between multiple experiments (batch effects) and a high number of technically absent expression values (dropout). This can impede common analysis, for example, differential expression analysis, clustering and cell type identification. Common methods for scRNA-seq analyses focus on solving either the problems due to batch effects by batch correction or the problems of dropouts by imputation. However, both problems are closely related.

Given this insight, a combined approach for expression reconstruction, called DISCERN, was developed and extensively evaluated in the project described here. DISCERN, a generative deep learning model, is the first approach, which combines batch correction with imputation. It is based on the autoencoder architecture of Wasserstein autoencoders (WAEs) and conditional instance normalization (CIN) to reconstruct and adjust gene expression values to a reference batch.

DISCERN was extensively compared to previous batch correction and imputation methods. In several benchmarks, it outperforms state-of-the-art methods for batch correction, e.g. Seurat, scGEN, and scVI, as well as state-of-the-art imputation methods, e.g. scImpute, CarDEC, and DCA. The approach of DISCERN differs from previous approaches for batch correction and imputation by directly adjusting gene expression information and using a high-quality reference for the reconstruction of multiple batches. In contrast, established batch correction methods rely on an adjusted embedding of gene expression values and current methods for imputation are not evaluated for data sets composed of multiple batches. The evaluations show that DISCERN improves the analysis of scRNA-seq data with respect to the detection of marker genes and cell type identification, when using e.g. single-nuclei RNA-sequencing (snRNA-seq) or bulk RNA-sequencing (RNA-seq) data as a reference.

Especially bulk RNA-seq data obtained from cells sorted by type is well-suited as a reference, as it usually has almost no dropout of gene expression values, due to technical reasons. Applying DISCERN to a scRNA-seq data set and a bulk RNA-seq reference data set delivered novel insights into the development of severe lung damage in the coronavirus disease 2019 (COVID-19). These insights could be verified using other data modalities.

Thus, reference-based reconstruction based on deep generative networks, such as the one implemented in DISCERN, provides a real advance in the analysis of Omics data.

Deutsche Zusammenfassung

Einzelzellsequenzierung (scRNA-seq) bietet detaillierte Einblicke in die Biologie der Gewebe- und Krankheitsentwicklung auf Einzelzellebene. Diese zellspezifischen Informationen können für die Zellidentifizierung, die Zellentwicklung und die Charakterisierung von Krankheiten genutzt werden. Die derzeitigen Sequenzierungsmethoden leiden jedoch unter technischen Einschränkungen, vor allem unter starken Unterschieden zwischen mehreren Experimenten (Englisch: Batch Effekte) und einer hohen Zahl technisch nicht vorhandener Expressionswerte (Englisch: Dropout). Dies kann Analysen kombinierter Datensätze, z. B. die Identifizierung von Zellgruppen und Zelltypen, behindern.

Gängige Methoden für die scRNA-seq-Analyse konzentrieren sich entweder auf die Lösung des Problems der Batch-Effekte durch Batch-Korrektur oder auf die Lösung des Problems der Dropouts durch Vorhersage der fehlenden Expressionswerte (Englisch: Imputation). Beide Probleme sind jedoch eng miteinander verbunden.

Daher wurde in dieser Arbeit mit DISCERN ein kombinierter Ansatz zur Expressionsrekonstruktion entwickelt und umfassend evaluiert. DISCERN ist ein generatives Deep-Learning-Modell und der erste Ansatz, der Batch-Korrektur mit der Vorhersage von fehlenden Expressionswerten kombiniert. Es basiert auf der Wasserstein-Autoencoder (WAE) Architektur und auf konditioneller Normalisierung (CIN), um Genexpressionswerte zu rekonstruieren und an eine Referenz anzupassen. Dieser kombinierte Ansatz liefert in verschiedensten Benchmarks bessere Ergebnisse als die modernsten Methoden zur Batch-Korrektur, z.B. Seurat, scGEN und scVI, und zur Imputation, z.B. scImpute, CarDEC und DCA.

Der Ansatz von DISCERN unterscheidet sich von früheren Ansätzen der Batch-Korrektur und Imputation durch das direkte Anpassen von Genexpressionsinformationen und durch Verwendung einer qualitativ hochwertigen Referenz zur gleichzeitigen Rekonstruktion mehrerer Experimente. Aktuelle Verfahren zur Batch-Korrektur beruhen hingegen auf der Anpassung einer Einbettung der Expressionsdaten und Methoden zur Imputation wurden nicht an Datensätzen getestet, die aus mehreren Experimenten bestehen.

Darüber hinaus kann DISCERN zur Verbesserung der scRNA-seq-Datenanalyse eingesetzt werden, z.B. zur Erkennung von zelltyp-spezifischer Genexpression und zur Identifizierung von Zelltypen. Dies wird anhand von scRNA-seq-, Einzelkernsequenzierung (snRNA-seq) und bulk RNA-seq-Datensätzen gezeigt. Insbesondere zelltypsortierte bulk RNA-seq Daten eignen sich sehr gut als Referenzdatensatz, da hierin nur wenige Expressionswerte aufgrund technischer Gründe fehlen. In einer mit DISCERN durchgeführt Analyse von scRNA-seq Daten mit bulk RNA-seq Referenzdaten ergaben sich neue Hypothesen über die Entwicklung schwerer Lungenschäden bei an COVID-19 erkrankten Personen. Diese Hypothesen wurde auf der Basis weiterer Datenmodalitäten verifiziert.

Die Ergebnisse dieser Arbeit zeigen, dass DISCERN mit der Kombination von referenzbasierter Rekonstruktion und tiefen generativen Modellen einen echten Fortschritt in der Analyse von Omics Daten darstellt.

Contents

1 Introduction

1.1 Single-cell RNA-sequencing

ScRNA-seq technologies measure gene expression at single-cell resolution, providing novel insights into the cellular composition and improving the understanding of cell-specific molecular processes [1, 2]. Several commercial platforms have facilitated researchers to use scRNA-seq methods at a reasonable cost. The main differences between these technologies include the use of droplet-based versus well-based cell capture, 3′ sequencing versus full-length sequencing, and the use or absence of unique molecular identifiers (UMIs). Well-based technologies, e.g. Smart-seq2, capture single cells in micro- or nanowells and perform the sequencing reactions inside these wells. This enhances the ability to detect gene expression with the cost of a lower number of sequenced cells [3, 4]. Droplet-based technologies, e.g. 10x Chromium, perform the sequencing reaction in oil-droplets achieving high throughput rates, but a lower number of detected genes per cell [3, 4]. Thus, well-based technologies usually provide a better gene-based characterization of cells, while droplet-based technologies provide more cells and thus enable the detection of rare cell types [5, 6]. Full-length sequencing provides information of whole transcripts and thus enables splicing analysis and gene variant detection at a single-cell level. When only transcript level expression is of interest, 3′ sequencing is preferred due to its lower costs [5]. UMIs are used to identify the initial RNA molecule, which is amplified using polymerase chain reaction (PCR) for sequencing. This enables the detection and removal of PCR amplification-related biases and therefore this is applied for most sequencing technologies if possible [7]. An overview of common sequencing technologies can be found in Table 1 and [8, 9]. Depending on the sequencing platform used, scRNA-seq technologies detect around 3 000 genes per cell, giving almost an order of a magnitude fewer genes detected than in bulk RNA-seq [3]. Despite major technological advances, the analysis of the

Table 1: Characteristics of common scRNA-seq technologies. Modified and extended from [8, 10].

Methods	Cell-Capture	Transcript coverage	UMI	Year	References
Tang method	well	Nearly full-length	No	2009	[11]
STRT-seq; STRT/C1	well	5′-only	Yes	2011	[12, 13]
CEL-seq	well	3′-only	Yes	2012	[14]
Smart-seq	well	Full-length	No	2012	[15]
Quartz-Seq	well	Full-length	No	2013	[16]
Smart-seq2	well	Full-length	No	2013	[17]
MARS-seq	well	3′-only	Yes	2014	[18]
Drop-seq	droplet	3′-only	Yes	2015	[19]
InDrop	droplet	3′-only	Yes	2015	[20]
CytoSeq	well	3′-only	Yes	2015	[21]
SUPeR-seq	well	Full-length	No	2015	[22]
CEL-seq2	well	3′-only	Yes	2016	[23]
Fluidigm C1	well	Full-length	No	2016	[24]
Chromium	droplet	3′-only	Yes	2017	[25]
DroNC-seq	droplet	3′-only	Yes	2017	[26]
sci-RNA-seq	well	3′-only	Yes	2017	[27]
Seq-Well	well	3′-only	Yes	2017	[28]
MATQ-seq	well	Full-length	Yes	2017	[29]
SPLiT-seq	well	3′-only	Yes	2018	[30]
Quartz-Seq2	well	3′-only	Yes	2018	[31]
DNBelab C4	droplet	3′-only	Yes	2019	[32]

high-dimensional scRNA-seq data remains one of the major challenges [33, 34]. Especially the sparsity of measured gene expression information and high technical noise impedes downstream analyses and thus represents one of the major technical downsides of single-cell sequencing. This is frequently called 'dropout' and refers to genes that are expressed by a cell but cannot be observed in the corresponding scRNA-seq data, which is a technical artifact. Dropout afflicts predominantly low to medium-expressed genes, as their transcript number is insufficient to reliably capture and amplify them. This missing expression information limits the resolution of downstream analyses, such as cell clustering, differential expression, marker gene detection, and cell type identification [35]. Recent studies suggest that the number of technical dropout in UMI-based sequencing technologies is lower than expected and most absent expression values have biological reasons [36, 37]. Furthermore, statistical models for imputation of absent expression values relying on zero-inflated distributions or removing dropout may introduce more noise than signal [37]. However, as Jiang *et al.* [37] discussed, the true effect of biological and non-biological missing expression information and their effect on downstream analyses is still an open question. To cope with the missing gene expression information in single-cell experiments, several *in silico* gene imputation methods have been designed. Gene imputation infers gene expression in a given cell type or state, based on the information from other biologically similar cells of the same data set. Several methods utilizing this principle have been developed [38], amongst them, DCA [39], MAGIC [40], scImpute [41], DeepImpute [42], and CarDEC [43]. An important prerequisite for applying these imputation methods is an appropriate similarity measure between cells (and their gene expression profile). The systematic development of such similarity measures is an unsolved problem [44]. Thus, most imputation methods providing improved gene expression information rely on the comparison of similar cells with largely absent gene expression information. Genes that are not expressed in neighboring cells cannot be imputed, limiting the application of scRNA-seq imputation as described above. In an ideal case, it would be possible to obtain information on the true gene expression per cell, or at least expression information with less technical noise, to reconstruct the true expression at the single-cell level. However, this information is available only for very few single-cell studies. These studies are usually used for benchmark purposes only, for example to compare scRNA-seq and snRNA-seq [45]. Furthermore, the basis for several of these algorithms is a zero-inflated distribution, which is potentially inappropriate

[37]. Another class of imputation algorithms uses bulk RNA-seq data to constrain scRNA-seq expression imputation, for example, Bfimpute [46], SCRABBLE [47] and SIMPLEs [48]. These methods usually require bulk RNA-seq data from the same or similar tissue and with the same cell composition [47]. This additional information is used to estimate the true gene expression of the single-cell data set constrained by the bulk RNA-seq data and to fill in missing values. Beyond dropout, there are other technology and data set-specific changes of the expression profile, e.g. capture rate of specific genes and differences in sample processing, which affect the single-cell data analysis. These changes can be classified into wanted (enforced by the experimental setup) or unwanted (stochastic changes in the experimental setup, material) changes [49]. All these technology and data set-specific differences, including dropout, are usually referred to as batch effects [49]. Common sources of batch effects include experimental design, time points of extraction of the biological material, material handling, operators, reagent quality, equipment, library preparation, and the sequencing technology, as discussed above [49–51]. Already slight and unwanted variations of these and other experimental conditions can induce batch effects. Furthermore, scRNA-seq data sets are often combined between experiments or with publicly available data, which makes an identical sample handling nearly infeasible [49, 51]. Correction of batch effects without considering the experimental setup can lead to unwanted removal of wanted experimental differences [51, 52]. The exact source of batch effects is often unknown [51]. Recently, automated methods for the detection of batch effects have been developed [51], but these still can only provide limited information on the source of the batch effect. In the following, the term batch will refer to one single-cell experiment, whereas a data set consists of one or more simultaneously analyzed single-cell experiments, with one or more batches.

1.2 Expression reconstruction methods

The problem of expression reconstruction consists of generating missing expression values and adjusting measured expression data to achieve a better quality of the data set. Thus, in the following expression reconstruction will be used as a synonym for batch correction in combination with imputation, yielding reconstructed or corrected data. Reconstructed and corrected data is used synonymously.

There are many different methods to solve this problem, see Table 2 for a list of methods. These methods can roughly be divided into imputation-focused and batch-correction-focused. Imputation-focused methods aim to reconstruct missing expression values from other cells in the same data set. They usually do not perform adjustments of measured expression data, i.e. non-zero values. Batch-correction-focused methods aim to remove the batch-specific differences. These methods were initially developed to remove only the batch effect from lower dimensional representations [50, 53, 57], but recently developed approaches additionally remove batch effects on the gene expression level [43, 54, 55]. However, those methods often include their own algorithms for downstream application on the gene level, e.g. differential gene expression testing in scVI [55] or were not evaluated for gene expression imputation, e.g. Seurat [53].

1.2.1 Imputation-focused methods

Markov affinity-based graph imputation of cells (MAGIC) [41] uses data diffusion-based information sharing for imputation and denoising of scRNA-seq count matrices. However, the construction of an appropriate similarity metric is challenging, but necessary for imputation [44]. Thus, MAGIC uses a graph-based approach that builds less noisy cell-cell affinities and uses information sharing across cells to measure cell-cell similarity. MAGIC particularly focuses on the understanding of gene-gene relationships to better characterize dynamics in biological systems. The tool is provided as a Python package.

Deep Count Autoencoder (DCA) [39] uses an autoencoder based method (see Section 1.4.2 for details on autoencoders) for denoising scRNA-seq count matrices. The Zero-inflated negative Binomial (ZINB) distribution is used to model the expression data and the high dropout rate of scRNA-seq data. DCA tries to estimate gene-specific parameters of the ZINB distribution, namely dropout, dispersion and mean. Using the estimated distributions as a noise model, DCA can compute the dropout probabilities of each gene and denoise and impute the missing counts by identifying and correcting dropout events. DCA is implemented in Python and TensorFlow.

Table 2: Overview of common methods for expression reconstruction.

Method	Use case	Gene space	Deep learning	Data modality	Other use cases	References
MAGIC	Imputation	✓		scRNA-seq		[41]
DCA	Imputation	✓	✓	scRNA-seq		[39]
scImpute	Imputation	✓		scRNA-seq		[40]
DeepImpute	Imputation	✓	✓	scRNA-seq		[42]
Seurat	Batch correction	(partial)		scRNA-seq		[53]
scGen	Batch correction	✓	✓	scRNA-seq	perturbation prediction	[54]
scVI	Batch correction	✓	✓	scRNA-seq	clustering, differential expression	[55]
CarDEC	Imputation + batch correction	✓	✓	scRNA-seq		[43]
trVAE	Batch correction	✓	✓	scRNA-seq	prediction of unseen events	[54]
Bfimpute	Imputation	✓		scRNA-seq + bulk RNA-seq		[46]
SIMPLEs	Imputation	✓		scRNA-seq + bulk RNA-seq		[48]
SCRABBLE	Imputation	✓		scRNA-seq + bulk RNA-seq		[56]

scImpute [40] is a method using a three-step procedure for the imputation of scRNA-seq data. The first step consists of dimensionality reduction using Principle Component Analysis (PCA) and spectral clustering to detect groups of similar cells, which are handled by the model separately. For each group of cells, scImpute fits a mixture model of gamma distributions and a normal distribution to distinguish technical and biological dropout in the second step. Finally, a cell-specific regression model is used for the imputation of genes with a high probability of dropout. This approach is used to prevent hallucinations and maintains the gene expression distribution. scImpute is provided as an R package.

DeepImpute [42] uses an ensemble of multiple autoencoder-like deep neural networks. The gene expression matrix is split into multiple subsets of input and target genes. Each network is trained to learn the gene-gene relationship between a set of input genes and some target genes. Input and target gene sets are selected based on the correlation of gene expression values. The estimated expression values from each of the networks are combined to yield the final imputed data set. DeepImpute is implemented in Python and TensorFlow.

SIMPLEs [48] is a statistical framework for imputing scRNA-seq data using bulk RNA-seq data. scRNA-seq expression data is modeled using mixtures of zero-inflated censored multivariate Gaussian distributions. With available bulk RNA-seq data and the assumption of no dropout in bulk RNA-seq, SIMPLEs estimates a gene-specific dropout rate per cell type. Thus, SIMPLEs requires bulk RNA-seq data with the same cell types as in the scRNA-seq data. SIMPLEs is implemented in R.

Bfimpute [46] uses Bayesian Probabilistic Matrix Factorization to decompose scRNA-seq data into a latent cell and a latent gene matrix. The resulting model is then used in further steps of the Markov Chain Monte Carlo algorithm to get the estimated dropout rates. Cell-specific information or gene-specific information, i.e. bulk RNA-seq data can be used for adjusting the Gaussian prior distribution before applying probabilistic matrix factorization. Bfimpute is implemented in R.

SCRABBLE [47] is based on matrix regularization and operates on scRNA-seq and bulk RNA-seq data. It tries to impute the scRNA-seq data using three principles, implemented in loss functions. First, the imputed scRNA-seq expression matrix should be close to the real scRNA-seq expression matrix. Second, the rank of the imputed scRNA-seq should be as small as possible, since only a limited number of clusters (and cell types) are present in a data set. Third, the loss function operates on the bulk RNA-seq data and tries to minimize the distance between the average imputed scRNA-seq data and the (average) bulk RNA-seq data. SCRABBLE is implemented in R.

1.2.2 Batch-correction-focused methods

Seurat [53] was implemented as an open-source toolkit for the analysis of scRNA-seq data, where batch-correction functionality is included. Seurat uses either Canonical Correlation Analysis (CCA) or reciprocal PCA. In case it uses CCA, the cells from different batches are embedded in a common space using Singular Value Decomposition (SVD) on the cell-cell-correlation matrix. In case reciprocal PCA is used, one batch is embedded into the PCA space of another batch. In this lower dimensional representation, Seurat tries to find neighbors between batches, called anchors. Anchors are filtered considering the local neighborhood of the cell pairs and remaining anchors used to construct correction vectors for all cells. This enables batch correction in the lower dimensional representation, but due to the ability to reverse CCA and PCA, expression information can also be reconstructed. Seurat is provided as an R package.

scGen [54] uses a variational autoencoder (VAE) (see Section 1.4.3 for more details) to embed scRNA-seq data with different conditions into the same lower dimensional representation. This common representation is then used to estimate perturbation vectors, e.g. for cells reacting to a drug treatment. These perturbation vectors can be used to predict a (drug treatment) effect, which was not observed. These linear perturbation vectors can also be used for the correction of batch effects. Therefore, scGen uses both batch and cell type labels to overcome cell type-specific batch effects. scGen is built using scvi-tools [55] and implemented in Python and PyTorch.

scVI [58] is a probabilistic model for scRNA-seq data analysis including batch correction, clustering, and probabilistic differential expression analysis. It is based on the VAE framework and models the expression data using the ZINB distribution and a corresponding loss function. The use of a parametric model in scVI allows for the inclusion of confounding variables, e.g. batch annotation, and their correction. scVI is part of the scvi-tools [55] toolbox and is implemented in Python and PyTorch.

Count adapted regularized Deep Embedded Clustering (CarDEC) [43] is a method for batch effect correction, denoising of expression data, and cell clustering. It is based on the autoencoder framework and imputes the expression matrix in two steps. In the first step highly variable genes across all batches are used to pretrain an autoencoder. In the second step, the learned weights are transferred to a network for modeling high and low variable genes using two reconstruction losses. One loss is computed for the highly variable genes and another for the low variable genes. Additionally, a self-supervised clustering loss in the latent space is included to improve batch mixing. CarDEC is implemented in Python and TensorFlow.

transfer VAE (trVAE) [54] uses a variational autoencoder for out-of-distribution data generation. It can generate unseen samples or conditions of scRNA-seq data by encoding the expression matrix together with an additional input for the condition. The decoder is then trained to reconstruct the encoding to a specified target condition. To achieve condition independence the first layer of the decoder is regularized using maximum mean discrepancy. This framework can therefore also be used for the correction of batch effects by treating the batch of origin as a condition label. trVAE is implemented in Python and TensorFlow.

All above mentioned methods were developed only for batch correction or imputation, however, both tasks show a strong connection. For example, theoretically imputed expression data yielding biological ground truth, should not show any technical batch effect. Imputation tools are often only tested on single-experiment data sets whereas batch correction methods are not tested for imputation of gene expression information. Therefore, adaptation of downstream applications to varying number of absent expression values and multiple batches is often necessary, which impedes downstream applications.

1.3 Evaluation methods / Downstream applications

1.3.1 Evaluation of Imputation

The evaluation of imputation algorithms can be performed directly on the level of gene expression or on downstream applications. The evaluation of gene expression can be done either by considering the mean expression per cell type or by single expression values in the case of simulated data. For downstream applications, usually, clustering, which will be discussed in Section 1.3.2, differential gene expression analysis, and pathway enrichment analysis [38] are applied. These techniques will be used to evaluate the performance of all methods in Chapter 3.

(Mean) gene expression Comparison of gene expression values can be done using metrics such as the mean squared error (MSE) or correlative measurements including Pearson or Spearman's rank correlation. However, for comparing methods on multiple data sets Pearson correlation has the advantage, that it is independent of the scale of the gene expression values and the number of samples, i.e. in MSE larger values in the comparison usually lead to a larger MSE. If no ground truth gene expression information per cell is known, comparing the average gene expression per cell type can be an advantage, because only a cell type-specific and no cell-specific reference is needed. Furthermore, the use of Pearson correlation is advantageous over rank-based correlation methods, since in scRNA-seq expression data many data points have values close to zero, which would be considered as equal weights in rank-based methods. In the case of comparing mean expression values, stratification by cell type is important since the gene expression can be largely different by cell type.

Differential gene expression analysis On a higher level the detection of marker genes, using differential expression analysis (DEA) [59], can be used to evaluate the performance of imputation algorithms. This is usually done by performing t-tests for each gene comparing all cells of a cell type/cluster against all other cells in the data set together with computing a $\log_2$-fold-change of the mean gene expression. This can be compared to the same evaluation, i.e. t-test as described above, on ground truth data. For comparison, either the $\log_2$-fold-change, the t-statistics, or the p-value from significance testing (t-test) can be used. Since the p-value is a probability, expressing the significance of a value, it does not reflect the magnitude of the change and so correlative comparisons on p-values can barely be used. In contrast, the $\log_2$-fold-change indicates the magnitude of a change, but does not reflect the significance of a value. Furthermore, especially low expressed genes, with expression values close to zero, can generate large, but insignificant $\log_2$-fold-changes. Therefore, no evaluations on p-values and a limited number of evaluations based on $\log_2$-fold-changes are performed.

The t-statistics for a gene is defined as $t = \sqrt{n}\frac{\mu_1 - \mu_2}{\sigma}$, where μ_1 is the mean of group one (e.g. a cell type) and μ_2 is the mean of a group two (e.g. another cell type, all other cells), σ is the standard deviation of the expression value of all cells and n is the total number of cells in group one and group two. Therefore the t-statistics summarizes the quantity of the effect ($\mu_1 - \mu_2$) and the variation σ. Thus, comparisons based on the t-statistics enable the simultaneous comparison of the effect strength and the variation using a single value for each gene. They are therefore preferred over comparisons based on $\log_2$-fold-changes and p-values. The comparison of t-statistics can be performed using any correlative measurement, e.g. Pearson or Spearman's rank correlation.

Gene set enrichment analysis is a method for detecting over-representation of *a priori* defined gene sets in the top or bottom of a ranked list of genes [60]. This method can be used to detect a correlation between the expression of gene sets with a phenotype by using DEA results to rank genes. Several gene sets summarizing experimentally proven cellular pathways or common cellular functions are available [61]. The most common databases are the gene ontology (GO) database [62] and the Kyoto Encyclopedia of Genes and Genomes (KEGG) [63] database. The ranking of the experimentally generated gene lists using differential gene expression analysis can be done using statistics from DEA, e.g. t-statistics, or the $\log_2$-fold-change. Due to their limited range, p-values are usually not used for ranking.

1.3.2 Evaluation of batch correction

Dimensionality reduction and visualization t-distributed stochastic neighbor embedding (t-SNE) [64] and Uniform Manifold Approximation and Projection (UMAP) [65] can be used to visualize scRNA-seq data sets and to qualitatively assess integration performance. Both methods are usually based on principle components from PCA and use non-linear functions, described for each method below, to create a two-dimensional representation of the data [66]. These embeddings should represent the local structure (local similarity of cells) and the global structure of the high dimensional scRNA-seq data well.

t-SNE consists of a two-step procedure. In the first step, a probability distribution P of pairs of points is computed, such that similar points have a high probability of pairing and dissimilar point pairs get a low probability assigned. The similarity of two pairs of points x_i and x_j can be computed using $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$, where N is the total number of points and

$$p_{i|j} = \begin{cases} 0 & \text{if } i = j \\ \dfrac{\exp(-|x_i - x_j|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-|x_i - x_k|^2/2\sigma_i^2)} & \text{otherwise.} \end{cases}$$

with a data set and point-specific parameter σ_i for each point x_i. σ_i can be estimated using a user-defined value called perplexity and a binary search [64]. By optimizing the Kullback-Leibler Divergence $\mathrm{KL}\,(P \parallel Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$ [67] between this distribution P and a similar distribution in the t-SNE embedding Q, an optimal localization of the points can be found to represent the data points in 2D [64]. Q herein measures the similarity in the embedding by computing

$$q_{ij} = \begin{cases} 0 & \text{if } i = j \\ \dfrac{(1 + |y_i - y_j|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + |y_k - y_l|^2)^{-1}} & \text{otherwise.} \end{cases}$$

where y are the coordinates in the t-SNE embedding. t-SNE is focused on capturing local similarity and therefore sometimes loses global structure [66].

UMAP follows a similar approach by constructing a similarity graph in high dimensional space. The network edges are weighted by the distance between points. Similar to t-SNE, in the second step the construction of a graph in the lower dimensional embedding is performed, which is optimized to be similar to the similarity graph in high dimensional space [65]. Furthermore, UMAP was developed to capture the global structure of the data better than t-SNE [68]. However, it is still discussed in recent literature [69], whether this development goal is achieved by UMAP.

Silhouette score [70] is a measure to evaluate clustering performance by comparing the mean intra-cluster distance to the mean nearest-cluster distance. The Silhouette score can be computed for batch and cell-type labels. The score has values in the interval $[-1, 1]$, such that a positive value indicates separated clusters, a value of zero signifies cluster overlap, and a negative value represents the case that for all clusters the closest cluster is the wrong cluster. For evaluating batch mixing, i.e. overlap of batches, a low, close to zero, value is best, while for evaluations compared to cell type clusters, i.e. preservation of biology, a value close to 1 is best. Conservation of cell type clusters (high Silhouette scores when comparing to cell type labels) indicate that for the cells the correct cellular identity is kept and thus the underlying biology in the data set is preserved.

For a cluster set $U = \{U_1, U_2, \ldots, U_n\}$ of non-overlapping sets U_i with $U_i \cap U_j = \emptyset \ \forall \ i \neq j$ and all elements $c \in U_i$ having a unique cluster label i, the Silhouette score is defined as:

$$\mathrm{S}(U) = \frac{1}{\|U\|} \sum_{i=1}^{n} \begin{cases} 0 & \text{if } |U_i| = 1 \\ \sum_{c \in U_i} \frac{\mathrm{dist}(c, U_{\neq i}) - \mathrm{dist}(U_i, c)}{\max(\mathrm{dist}(U_i, c), \mathrm{dist}(c, U_{\neq i}))} & \text{otherwise.} \end{cases} \tag{1.1}$$

where $\|U\|$ is the total number of elements in the cluster set, n is the number of sets in U, $\mathrm{dist}(c, U_i)$ is the mean distance of an element c to all elements of cluster U_i and $\mathrm{dist}(c, U_{\neq i}) = \min_{j \neq i} \mathrm{dist}(c, U_j)$ is the smallest mean distance of an element c to any cluster, which is not U_i.

Adjusted Rand index [71] The Rand index estimates the similarity between two cluster sets by comparing all possible pairings of samples to estimate how often samples share the same cluster in the two cluster sets. The adjusted Rand index (ARI) is normalized, such that a random labeling will result in a value close to 0, while a perfect clustering yields a score of 1. The ARI is computed on the result of clustering algorithms.

For two cluster sets $U = \{U_1, U_2, \ldots, U_n\})$ and $V = \{V_1, V_2, \ldots, V_m\})$ of the same same elements $\cup_i U_i = \cup_j V_j$ the ARI is defined as:

$$\mathrm{ARI}(U, V) = \frac{\sum_i \sum_j \binom{|U_i \cap V_j|}{2} - \left(\sum_i \binom{|U_i|}{2} \sum_j \binom{|V_j|}{2}\right) \Big/ \binom{N}{2}}{\frac{1}{2}\left(\sum_i \binom{|U_i|}{2} + \sum_j \binom{|V_j|}{2}\right) - \left(\sum_i \binom{|U_i|}{2} \sum_j \binom{|V_j|}{2}\right) \Big/ \binom{N}{2}} \tag{1.2}$$

where $N = \|U\| = \|V\|$ is the total number of elements.

Adjusted mutual information [72]

Mutual information measures the similarity between two cluster sets by computing the sizes of the intersection of all possible cluster label pairs. The adjusted mutual information (AMI) is adjusted for chance, such that a random labeling will result in a value close to 0, while a perfect clustering yields a value of 1. Additionally, this accounts for the fact, that mutual information is generally higher for cluster sets with larger numbers of clusters. The AMI is computed on clustering results as described for the ARI. The AMI is defined as follows, using a similar notation as above:

$$\mathrm{AMI}(U,V) = \frac{\mathrm{MI}(U,V) - \mathbb{E}\{\mathrm{MI}(U,V)\}}{\max\{H(U), H(V)\} - \mathbb{E}\{\mathrm{MI}(U,V)\}} \tag{1.3}$$

where $H(U) = -\sum_i \frac{|U_i|}{\|U\|} \log \frac{|U_i|}{\|U\|}$ is the entropy of a cluster set U and the mutual information

$$\mathrm{MI}(U,V) = \sum_i \sum_j \frac{|U_i \cap V_j|}{N} \log \frac{|U_i \cap V_j| N}{|U_i||V_j|}$$

where $N = \|U\| = \|V\|$ is the total number of elements.

The expected mutual information of two random cluster sets is:

$$\begin{aligned} \mathbb{E}\{\mathrm{MI}(U,V)\} = & \sum_i \sum_j \sum_{z=\max(1,|U_i|+|V_j|-N)}^{\min(|U_i|,|V_j|)} \frac{z}{N} \log\left(\frac{N \cdot z}{|U_i||V_j|}\right) \\ & \times \frac{|U_i|!|V_j|!(N-|U_i|)!(N-|V_j|)!}{N!z!(|U_i|-|z)!(|V_j|-z)!(N-|U_i|-|V_j|+z)!} \end{aligned} \tag{1.4}$$

1.4 (Deep) Generative models

In machine learning, algorithms can roughly be grouped into discriminative and generative approaches. Discriminative models try to model the conditional probability $P(Y \mid X)$, i.e. trying to model the probability of an event Y given a state X, which is often observed data. In comparison, a generative model tries to model the joint probability $P(X, Y) = P(X \mid Y)P(Y)$, i.e. modeling the relationship of Y and X, allowing for data generation (state) given a event and vice versa [73].

Thus generative models provide a powerful tool for supervised as well as unsupervised generation and transformation of data [74]. The idea of generative models was adapted to deep neural networks as well. The most common deep generative models are generative adversarial networks and autoencoders. Later in Section 2.2 a WAE, which is a special kind of the autoencoder architecture, is used as the framework for modeling scRNA-seq data sets.

1.4.1 Generative Adversarial Networks (GAN)

In 2014, Goodfellow *et al.* proposed generative adversarial networks (GANs), which consist of a generator network and a discriminator network (Figure 1) [75]. The generator network is trained to generate data from a prior distribution, e.g. random data points. The discriminator network in contrast is trained to distinguish the data, generated by the generator, and real data. The generator network and the discriminator network are jointly trained in competitive, adversarial manner. Using this approach GANs were successfully used in robotics [76] and computer vision [77]. GANs were applied in many different biomedical applications to e.g. design small molecules [78], to generate artificial single-cell expression data [79, 80] and to impute data [81].

1.4.2 Autoencoder

In contrast to GANs, autoencoders are implemented by a single network, which consists of two parts: an encoder and a decoder (cf. Figure 2) [82]. The encoder $\mathcal{F}(X)$ is trained to encode the data into a representation of the data, which is later decoded by the decoder part (or generator) $\mathcal{G}(X)$ into the original representation [82]. Using this approach the network can be trained end-to-end by comparing the generated, decoded data with the original input data, i.e. for some input X and an arbitrary loss function $L(X, \hat{X})$ the model will be trained using $L(X, \mathcal{G}(\mathcal{F}(X)))$. To prevent learning the identity functions for an input matrix $X \in \mathbf{R}^{n \times m}$ $\mathcal{F}(X) = X$ and $\mathcal{G}(X) = X$ usually the output of the encoder part is (much) smaller than the input $\mathcal{F}(X) = Z$ with $Z \in \mathbf{R}^{n \times \ell}$ and $\ell \ll m$ [83]. These autoencoders are called undercomplete [84, 85]. In this case, the encoder is forced to encode the input data into an embedding, which has to contain sufficient information to (approximately) reconstruct the input data using the decoder [83]. Thus, autoencoders are a powerful technique to learn an embedding of the data. While autoencoders provide useful embeddings for existing data [83, 86], they are bad at generating new data or data interpolated from existing data [87]. Therefore, autoencoders were extended to probabilistic models. This idea was implemented in VAEs [88].

1.4.3 Variational Autoencoder (VAE)

VAEs were developed by Kingma & Welling in 2014 as the first probabilistic framework for autoencoders. Instead of encoding a data point x_i into an embedding z_i, the data point is encoded into a distribution $Q(Z|x_i)$ (Figure 3) [88, 89]. This is usually achieved by predicting a mean and variance estimate for each data point using the reparameterization trick [88], which allows for a probabilistic embedding while being differentiable. Having a model built upon differentiable functions only is important for neural networks, including autoencoders, such that they can be trained using backpropagation [90]. The reparameterization trick, including a Gaussian prior distribution, is achieved by modeling $Q(Z \mid x_i)$ using an estimate of the mean μ_i and an estimate of the variance σ_i for each data point, such that $z_i = \mu_i + \sigma_i \epsilon$ for a random variable $\epsilon \sim \mathcal{N}(0,1)$ sampled from a Gaussian distribution $\mathcal{N}(0,1)$ with zero mean and unit variance.

This reparameterization trick is used to reformulate the sampling process, such that the Kullback-Leibler Divergence [67] $D_{KL}\big(Q(Z \mid x_i) \parallel \mathcal{N}(0,1)\big)$ can be computed as a loss function between the estimated probability distribution $Q(Z \mid x_i)$ and the prior, a Gaussian distribution $P_Z = \mathcal{N}(0,1)$ [88]. Thus, VAEs try to match the estimated probability distribution $Q(Z \mid X = x_i)$ to the prior distribution P_Z of each data point x_i of X (Figure 4A) [88, 91].

Together with an appropriate reconstruction loss, measuring the difference between input and decoded data, e.g. by MSE or cross-entropy, minimizing this loss function is equivalent to maximizing the Evidence lower bound (ELBO) [88]. The ELBO is a lower bound of the log-likelihood of the observed data with a given generative model [88]. VAEs tend to create representations, where individual data points cannot be clearly distinguished, e.g. blurry images [84]. However, this is not only the case for VAEs, but a common problem with generative models optimized using a log-likelihood [84, 92].

Despite these problems, VAEs are widely applied in the research of scRNA-seq experiments:

- Gayoso *et al.* build a framework for probabilistic analysis of single-cell data based on VAEs [55],
- Xu *et al.* used VAEs for data integration and cell type annotation transfer [93],
- Lotfollahi, Wolf & Theis predicted unknown drug perturbations using VAEs [94],
- Lotfollahi, Litinetskaya & Theis used VAEs for multimodal data integration [95] and
- Schriever & Kostka created a VAE based model for doublet prediction in single-cell experiments [96].

1.4.4 Wasserstein-Autoencoder

More recently, ideas from optimal transport [97] were applied to autoencoders and GANs [91, 98]. These networks were designed to explicitly minimize the (Wasserstein, or earth-mover) distance between the distribution of the input data and their reconstruction [99, 100]. The Wasserstein distance measures the distance between two probability distributions and can intuitively be seen as the amount of "mass" needed to be moved to convert between the two distributions. The problem of computing the optimal transport between two probability distributions was first formally described by Monge in 1781 [101]. Tolstikhin *et al.* used the idea of the Wasserstein distance to create an autoencoder architecture, which minimizes the Wasserstein distance, called WAE. Compared to VAEs, the WAE framework can use a wide range of architecture and losses [91]. In contrast to VAEs (Figure 4A), WAEs are trained to match the prior distribution P_Z with the aggregated posterior distribution $Q_Z := \int Q(Z \mid X) dP_X$ (Figure 4B).

Using any non-negative cost function $c(x, y)\colon \mathcal{X} \times \mathcal{X} \to \mathcal{R}_+$, an arbitrary divergence $\mathcal{D}_Z$ between the aggregated posterior distribution Q_Z and the prior distribution P_Z and exploiting the Kantorovich-Rubinstein duality [91] the WAE aims to minimize the following objective function:

$$D_{\mathrm{WAE}}(P_X, P_G) := \inf_{Q(Z|X) \in \mathcal{Q}} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)} \big[c\big(X, G(Z)\big) \big] + \lambda \cdot \mathcal{D}_Z(Q_Z, P_Z), \tag{1.5}$$

where $\mathcal{Q}$ is a set of probabilistic encoders and $\lambda > 0$ is a hyperparameter. As usual, P_X denotes the distribution of the input data and P_G the distribution produced by the generator. Tolstikhin *et al.* proposed to use the Maximum Mean Discrepancy (MMD) [102] as a divergence measurement between the aggregate posterior Q_Z and the prior distribution P_Z, which can e.g. be a Standard Gaussian distribution [91]. Using a positive-definite reproducing kernel $k \colon \mathcal{Z} \times \mathcal{Z} \to \mathcal{R}$ and a corresponding real-valued reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ [103], the MMD can be defined as [102]:

$$\mathcal{D}_Z(P_Z, Q_Z) = \| \int_{\mathcal{Z}} k(z, \cdot) dP_Z(z) - \int_{\mathcal{Z}} k(z, \cdot) dQ_Z(z) \|_{\mathcal{H}_k} \tag{1.6}$$

The MMD can be interpreted as distance of the embedding of P_Z and Q_Z in an RKHS [102, 103]. For example, the sum over an inverse multiquadratic kernel with different sizes can be used as a divergence measurement [91]. Since in the WAE framework the aggregated posterior distribution is compared to the prior distribution, it allows the use of probabilistic encoders (as in the VAEs and Figure 3) or deterministic encoders (as in classical autoencoders and Figure 2). Rubenstein, Schoelkopf & Tolstikhin showed, that probabilistic encoders can be beneficial for WAEs, especially if the intrinsic dimensionality of the data is not known [104]. When using a probabilistic encoder the same reparameterization trick is applied as for VAEs (Section 1.4.3). The probabilistic encoder enables the WAE to fill the dimensions of the embedding with noise instead of spreading the encoded data to all components of the embedding [104]. These noise-filled components can later be ignored by the decoder, whereas in the case of a deterministic encoder, all information is spread across all dimensions of the embedding.

A probabilistic encoder can still collapse to a deterministic one to best reconstruct the data and reduce the amount of noise for the reconstruction. In this case, the probabilistic encoder learned to embed each data point into a distribution with a variance of zero and is therefore equivalent to a deterministic encoder. This collapsing effect can be prevented by using a regularization term that enforces that some dimensions of the variance embedding are close to 1 and thus enables the network to fill superfluous dimensions with noise. According to [104] such a regularization can be defined as follows:

$$S_\sigma(x) = \sum_{i=1}^{d_\mathcal{Z}} \left\| \log\left(\sigma_i^2(x)\right) \right\| \tag{1.7}$$

where d_z is the number of latent dimensions, $\|x\|$ is the absolute value of x and σ is the function generating the components of the variance in the latent space, e.g. the encoder network, and σ_i denotes the function generating only the i-th component of the variance.

Therefore, WAEs are a powerful technique for dimensionality reduction and embedding tasks, where the intrinsic dimensionality of the data is not known. Furthermore, WAEs are expected to achieve better reconstructions in comparison to VAEs, because each data point does not need to match the prior distribution (Figure 4) [91]. In contrast, in VAEs the estimated distributions tend to overlap to better match the prior distribution and therefore can lead to worse reconstructions in comparison with WAE.

While WAEs show these theoretical advantages, a thorough benchmark is still missing and VAEs remain the most widely used method, especially for scRNA-seq data analysis [55, 94, 105, 106]

1.5 Instance normalization

Instance normalization was developed to improve the performance of deep learning algorithms [107, 108]. For an single sample $x_i \in \mathbf{R}^m$ it can be defined as:

$$\mathrm{IN}_{\gamma,\beta}(x_i) = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{1.8}$$

for some mean estimate μ, variance estimate σ^2 and a small constant ϵ to avoid division by zero. Additionally, a scaling factor $\gamma \in \mathbf{R}^m$ and a shift factor $\beta \in \mathbf{R}^m$, where m is the number of genes, are learned during model training. These learnable parameters could revert the effect of the normalization, if this is beneficial for the performance of the network [107].

Equation (1.8) can be used for batch normalization (BN) [107] and layer normalization (LN) [108]. CIN (Section 1.5.3) [109], a modification of instance normalization, is later extended to continuous CIN to allow for projections between scRNA-seq data sets (see Section 2.1).

1.5.1 Batch normalization

Batch normalization (BN) was developed by Ioffe & Szegedy in 2015 [107]. In this context, the word "batch" does not refer to the data set, as in the sections before, but to the mini-batches used for training a deep learning model.

BN is often used for image recognition tasks using convolutional neural networks [110, 111]. It is also used in autoencoder networks for batch correction, e.g. scGEN [94] and scVI [55, 58]. The idea of BN is to achieve equal mean and variance for each mini-batch input during model training. For each input $x_i \in \mathbf{R}^m$, of a mini-batch $\mathcal{B}$ the mean $\mu_\mathcal{B}$ and variance $\sigma^2_\mathcal{B}$ can be computed as

$$\begin{aligned}\mu_\mathcal{B} &= \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} x_i \text{ and}\\ \sigma^2_\mathcal{B} &= \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}(x_i - \mu_\mathcal{B})^2.\end{aligned}$$

These mini-batch estimates of the population mean and variance can be used to shift and scale x_i to obtain a mean of zero and a variance of one (mean-variance scaling). As suggested in [107], Equation (1.8) can be modified to obtain BN as follows:

$$\mathrm{BN}_{\gamma,\beta}(x_i, \mathcal{B}) = \gamma\frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma^2_\mathcal{B} + \epsilon}} + \beta \tag{1.9}$$

During inference, $\mu_\mathcal{B}$ and $\sigma^2_\mathcal{B}$ can largely be different compared to training examples, if the mini-batch contains out-of-distribution samples. Thus, instead of relying on the estimates during inference, moving average estimates of $\mu_\mathcal{B}$ and $\sigma^2_\mathcal{B}$ from training are used [107]. Initially, BN was proposed to reduce the internal covariate shift, during the training of a deep neural network. The internal covariate shift refers to the phenomenon that for all i, $0 < i \leq n$ in a neural network $N = \{\ell_0, \ell_1, \dots, \ell_n\}$ and a given input x to the network, the input $\ell_{i-1}(\ell_{i-2}(\dots))$ to the inner layer ℓ_i changes during the training. This is due to the simultaneous training of previous layers which leads to adapted inputs to the considered layer. Careful selection of model and training parameters is necessary to ensure successful training [107]. Ioffe

& Szegedy [107] proposed that BN overcomes these problems by normalizing the input for each layer and therefore stabilize model training and reduce the training time.

However, Santurkar *et al.* showed evidence that this internal covariate shift is not the only reason for improved training performance of networks with BN [112]. Additionally, they propose that BN stabilizes and smooths the optimization landscape providing more efficient, predictive, and well-behaved gradients. These findings could explain other benefits of BN, e.g. the robustness to hyperparameter choice and fewer gradient explosion/vanishing problems in model training [112].

1.5.2 Layer normalization

Based on findings about BN, in 2016 layer normalization (LN) was proposed to further reduce the training time of networks trained on normalized batches and to simplify the use of normalization in recurrent neural networks [108]. BN, as described above, cannot be applied to recurrent neural networks and shows a strong dependency on the batch size [108]. Thus, instead LN is frequently used in recurrent neural networks for text generation [113] or protein structure prediction [114]. Instead of normalizing across all $|B|$ samples of a mini-batch, normalization is performed on a single sample x_i taking into account all m features, i.e. all inputs to a layer of the neural network. Therefore, sample mean μ_i and variance σ_i^2 are computed as

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x_{i,j} \text{ and}$$
$$\sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m}(x_{i,j} - \mu_i)^2$$

Thus Equation (1.8) can be formulated as suggested in [108]:

$$\mathrm{LN}_{\gamma,\beta}(x_i) = \gamma\frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta \tag{1.10}$$

When using LN, μ_i and σ_i^2 only depend on a single sample and thus, in contrast to BN, no moving averages are need, which needs to be updated in training and fixed during inference [108]. Thus, when using LN, Equation (1.10) can be applied in training and inference. It was shown that LN is especially beneficial for mini-batches of small sizes [108].

1.5.3 Conditional instance normalization

Instead of learning fixed values for the scaling factor γ and the shift factor and β as in BN (Equation (1.9)) and LN (Equation (1.10)), Dumoulin, Shlens & Kudlur proposed conditional scaling and conditional shift factors [109]. This approach is called CIN. It consists of modifying Equation (1.8) to learn scaling γ_s and shift factors β_s dependent on a condition s:

$$\mathrm{CIN}_{\gamma,\beta}(x_i, s) = \gamma_s \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta_s \tag{1.11}$$

When Equation (1.11) is used with mini-batch-wise estimates of $\mu = \mu_{\mathcal{B}}$ and $\sigma^2 = \sigma^2_{\mathcal{B}}$ as in BN it is called called conditional batch normalization (CBN). When Equation (1.11) is used with sample-wise estimates $\mu = \mu_i$ and $\sigma^2 = \sigma^2_i$ as in LN it is called conditional layer normalization (CLN).

By varying γ_s and β_s, it is possible to transfer the style of an artist to a new image [109]. Furthermore, a linear combination of multiple γ and β values from different conditions can be used to create multiple combinations of styles (Figure 5).

1.6 Research question

The removal of batch effects and imputation of technical missing values are challenging problems for scRNA-seq data analysis [37, 38, 51]. The large diversity of sequencing technologies is contributing to this problem by inducing technology-specific biases [8, 10]. Despite this limitation, data set-specific biases can be advantageous for data analysis, if properly exploited.

Generative deep learning models were successfully applied in modeling scRNA-seq data [55, 79, 80, 93–96]. They were also fruitfully used for style transfer in image data modeling [109]. Thus, the hypothesis is, that the combination of generative deep learning models (e.g. autoencoders) with style transfer techniques should allow for the transfer of the quality, considered as style of the scRNA-seq data, from high quality data sets to low quality data sets. This could improve several downstream analyses, for example, clustering, differential expression analysis, and cell type discovery. However, precise modeling of zeros in scRNA-seq experiments is important for several of these applications [115] and existing statistical models are potentially not appropriate [37]. Hence an investigation of distribution-free modeling of dropout values is needed.

This work will explore and evaluate the use of an autoencoder-based deep-learning model for improving scRNA-seq data analysis using reference-based expression reconstruction by exploiting and extending style transfer techniques.

While several methods were developed for either batch correction or imputation of scRNA-seq, the combination of these approaches with style transfer to use high-quality data as a reference represents a novel and flexible approach for scRNA-seq expression reconstruction. Furthermore, the newly developed tool and several state-of-the-art tools for batch correction and imputation are evaluated on many different data sets, covering a large range of sequencing technologies and tissue types. Therefore, several evaluation metrics are used to assess the performance in batch correction and imputation tasks on multiple aspects of scRNA-seq data analysis.

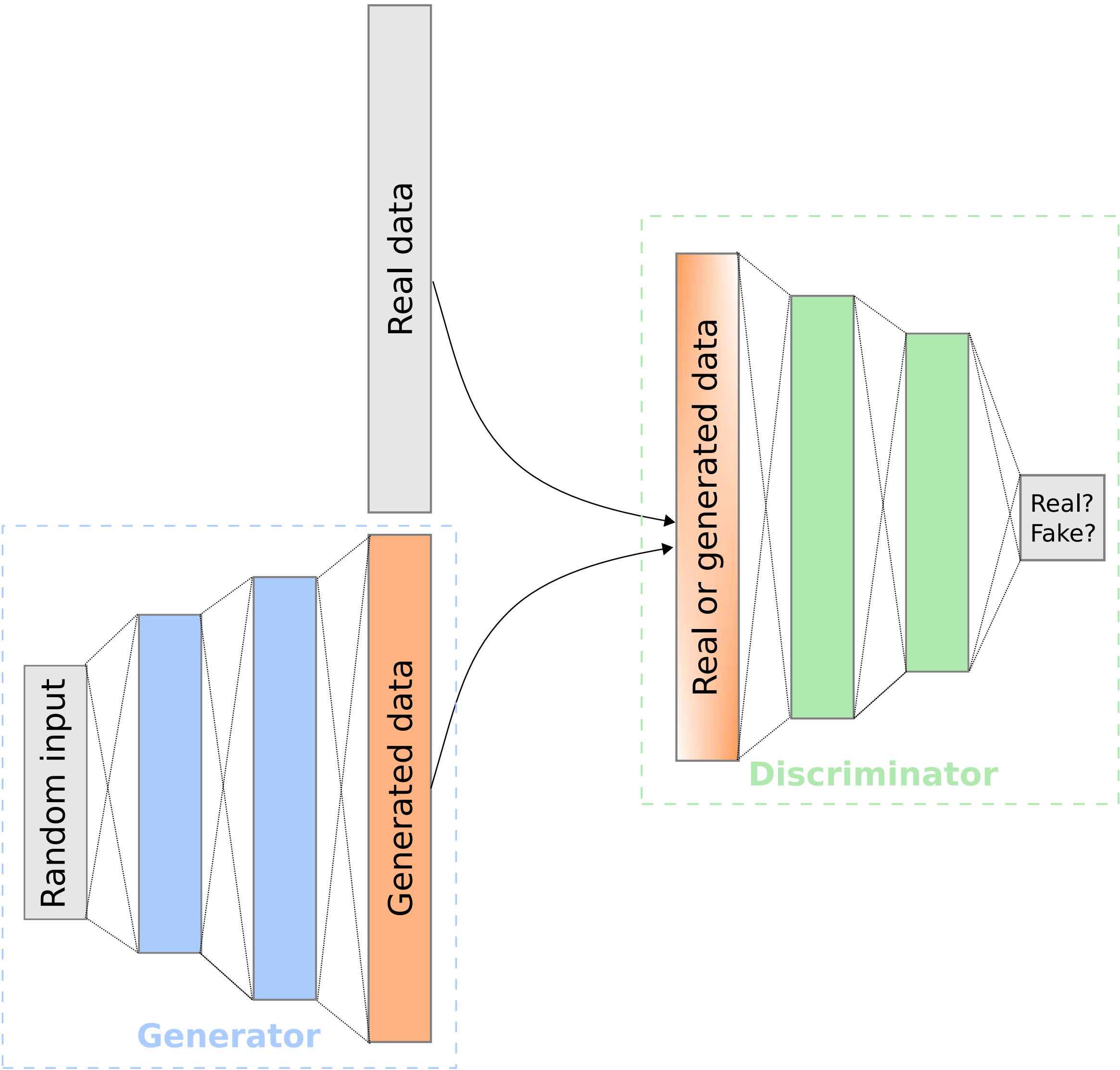


Figure 1: Schematic representation of a GAN. The network consists of a generator (blue) and a discriminator network (green), which are trained in an adversarial manner. The generator is trained to create realistic-looking data from random inputs. The discriminator is trained to distinguish real and generated (fake) data. Therefore the generator is trained to fool the discriminator.

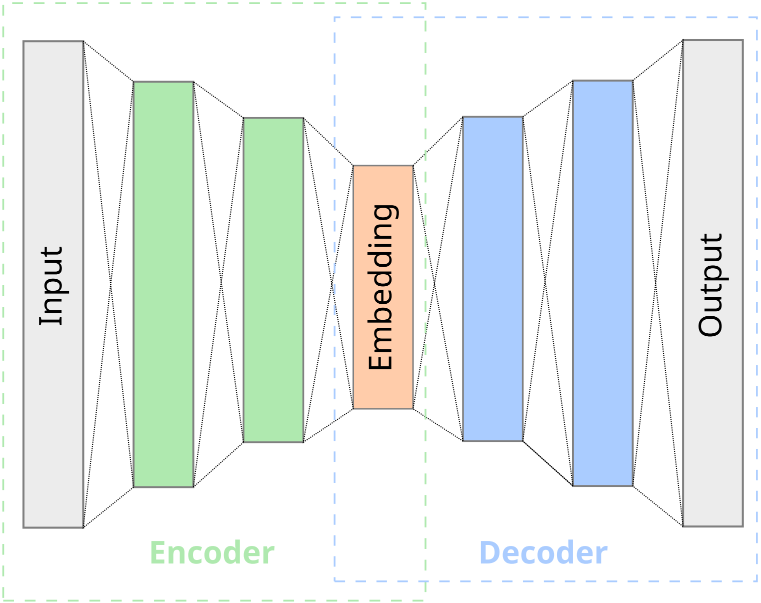


Figure 2: **Schematic representation of an autoencoder. The network can be separated into an encoder (green) and a decoder part (blue). The input is encoded into an embedding (orange) and decoded back to the output. The displayed network contains two hidden layers for both the encoder and the decoder.**

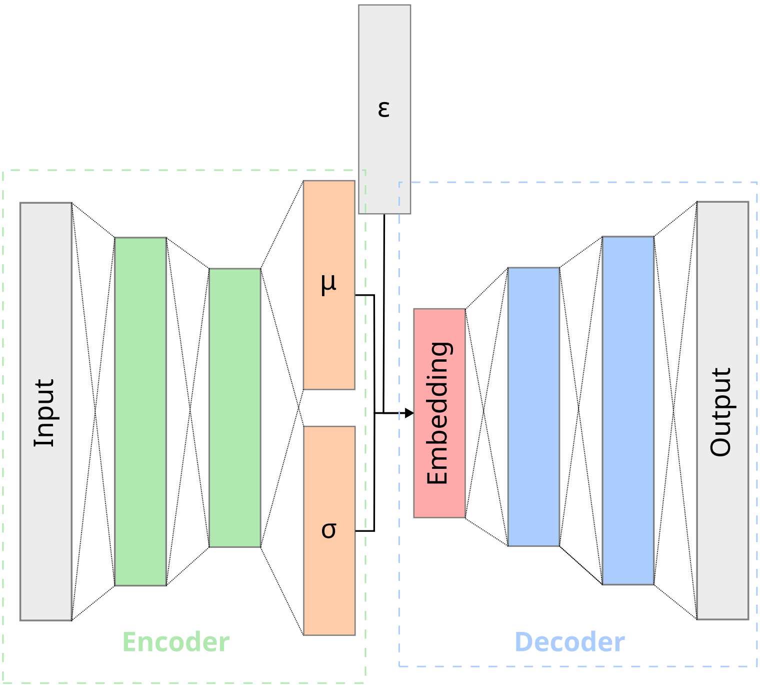
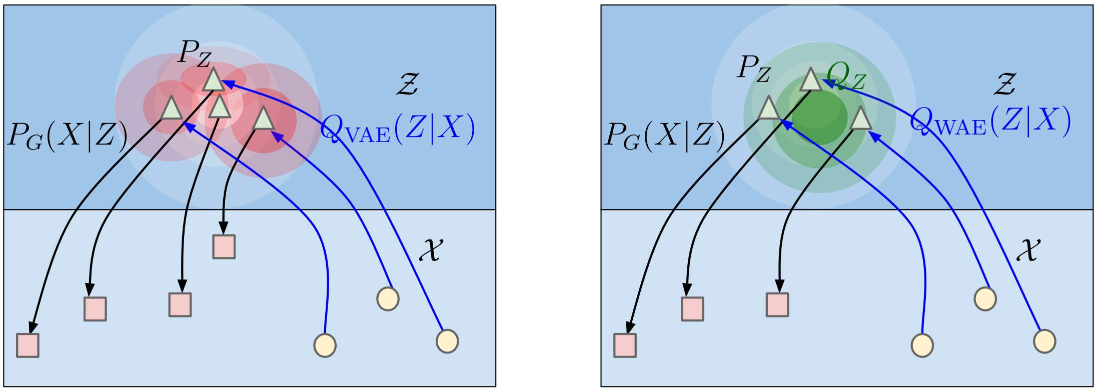


Figure 3: Schematic representation of a probabilistic autoencoder as used in VAEs. The network can be separated into an encoder (green) and a decoder part (blue). The input is encoded into a mean (μ) and variance estimate (σ) (orange). Using the reparameterization trick [88] and random noise (ϵ) the embedding (red) is randomly generated. This embedding can be decoded by the decoder (blue) back to the output. The displayed network contains two hidden layers for both the encoder and the decoder.



(A) VAE

(B) WAE

Figure 4: Graphical representation of (a) VAE based and (b) WAE based data modeling. The lower part (light blue) represents the input space X and the upper part (blue) indicates the embedding Z. The encoder part of the VAE and WAE respectively, is denoted using the probability function $Q_{\text{VAE/WAE}}(Z \mid X)$ and the decoder part with $P_G(X \mid Z)$. The prior distribution (P_Z) in the embedding is depicted by white circles. Input data points are shown as yellow circles, the embedding as light green triangles, and the reconstructions as red boxes. (a) In the VAE framework the estimated distribution for each data point, indicated in red, is compared to the prior distribution. (b) In the WAE framework the aggregated posterior distribution is compared to the prior distribution. This should prevent poor reconstructions due to overlapping points in the embedding. Adapted from [91].

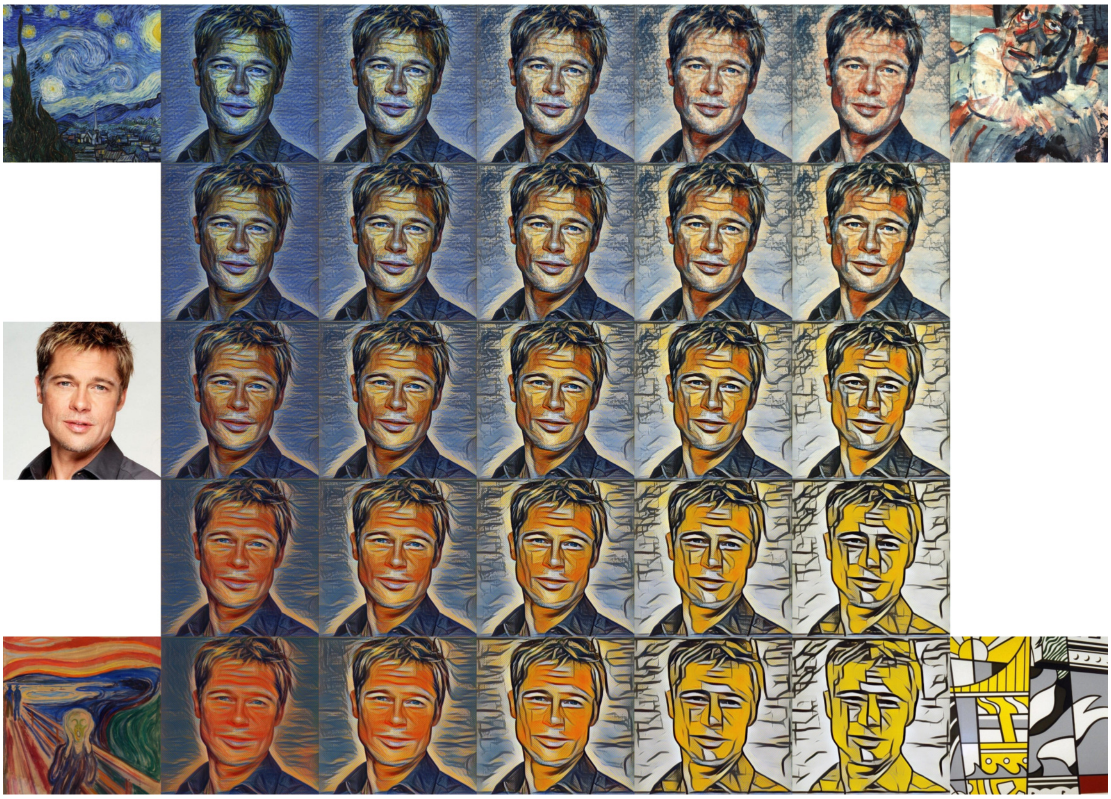


Figure 5: Style combinations from a style transfer network trained on 32 different styles. The combinations were obtained by conditional instance normalization. Four different styles, represented by the images in the corners, are applied to an image showing a person. The 32 images show varying degrees of style influence. Each image corresponds to a linear combination of γ and β values for the four different styles. The unmodified image of the person is shown on the left side. Image modified from [109].

2 Materials & Methods

Deep generative models are widely applied for imputation and batch correction tasks [39, 55, 58, 94]. When analyzing multiple data sets of different quality, it is not adequate to perform these steps one after the other. Instead, simultaneous batch correction and imputation are required to correct batch-specific dropout. So far no method has been developed that offers simultaneous batch correction and imputation of expression data for multiple data sets of different quality. The aim is to use a high-quality (hq) data set as a reference to reconstruct missing gene expression information in low-quality data (lq) data.

2.1 Continuous Conditional instance normalization

CIN was proposed for discrete conditions s, but was extended in this work to continuous variables (conditions) $v \in \mathbf{R}^m$, where m is the number of variables. For any sample $x \in \mathbf{R}^n$, scaling factor $\gamma \in \mathbf{R}^n$ and shift factor $\beta \in \mathbf{R}^n$ can be rewritten as $\gamma = W_\gamma v^T$ and $\beta = W_\beta v^T$ for $W_\gamma \in \mathbf{R}^{n \times m}$ and $W_\beta \in \mathbf{R}^{n \times m}$. W_γ and W_β are learned during model training, similar to β and γ in Equation (1.11). This modifies Equation (1.11) such that we obtain:

$$\mathrm{CIN}_{W_\gamma, W_\beta}(x, v) = W_\gamma v^T \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + W_\beta v^T \tag{2.1}$$

For a position (condition) s, $1 \leq s \leq n$, define a bitvector $v(s) = [v_1, v_2, \ldots, v_n]$ with $v_i = 1 \Leftrightarrow i = s$. Then, after replacing v by $v(s)$, Equation (2.1) reduces to Equation (1.11):

$$\mathrm{CIN}_{\gamma,\beta}(x, s) = \gamma_s \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta_s$$

where $\gamma_s = W_\gamma v(s)^T$ and $W_\beta v(s)^T = \beta_s$.

This reformulation allows continuous as well as one-hot-encoded[1] categorical variables to be used with CIN, e.g. any linear combination of categorical variables as in Section 1.5.3 and Figure 5 can be applied directly in model training or inference. To limit the expressivity of the conditioning, a constant scaling factor $W_\gamma v^T = 1$ can be beneficial (Equation (2.2)). If the scaling factor is set to 1, the network is limited to learning condition-specific shift factors only, formally expressed in the following equation.

$$\mathrm{CIN}_{W_\beta}(x, v) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + W_\beta v^T \tag{2.2}$$

2.2 DISCERN - Deep single-cell expression reconstruction

DISCERN, short for deep single-cell expression reconstruction, was developed to transfer the style from one scRNA-seq data set to another. This allows for transferring the data quality of a high-quality (hq) data set to a low-quality (lq) data set. Style transfer can be achieved using CIN and its extension proposed in Section 2.1. This approach combines batch correction and imputation to unify multiple scRNA-seq expression data sets and simplifies downstream applications, by removing batch effects from embeddings and from the expression data. Therefore, downstream applications do not need any further adaption for batch effects or varying numbers of absent expression values.

2.2.1 Architecture

DISCERNs architecture is based on the WAE framework (Section 1.4.4) consisting of three fully connected layers in the encoder and the decoder. An overview of the architecture is shown in Figure 6. Whereas the number of neurons for each layer can be also seen as a hyperparameter, the effect of it was not evaluated further. The number of neurons for all hidden layers were selected to powers of two to allow for potential running time improvements on CPU [117], which could be beneficial on GPU as well. The innermost layers, i.e. the lower dimensional representation or

[1] In one-hot encoding a category of n categories is transformed to a n-sized vector consisting of zeros except for a single one used uniquely to identify the category [116].

embedding, contain $n_{embed} = 48$ neurons, which gave most robust results according to the hyperparameter optimization described in in Chapter 3. However, to optimize the number of used dimensions for the embedding, the model contains a regularization term as proposed in Section 1.4.4 [104]. The regularization term is given in Equation (1.7). The MMD-penalty is used to compare the prior Gaussian distribution to the aggregated posterior distribution (Equation (1.6)). Besides the theoretical guarantees, as discussed in [91] and Section 1.4.4, this has the advantage that the model is enforced to create a more dense embedding compared to a classical autoencoder. This helps to remove batch effects in the embedding. Using the embedding produced by the random encoder, the decoder will try to reconstruct the original expression matrix. scRNA-seq data is known for a high level of zero measurements, i.e. dropout, which is essential to accurately represent the count data in modeling approaches. Despite the several non-linearities in the decoder architecture, modeling of multimodal distributions, e.g. gene expression data, using autoencoders is difficult [118]. For example, a the expression of a gene, which is highly expressed in one and not expressed in another cell type, could be modeled by a normal distribution, reaching a reasonable approximation in terms of the loss value. However, this is not biologically meaningful since all cell types would have the same mean expression levels and cell type specific difference were lost.

A common method to overcome this is to model the expression data using a parametric distribution, e.g. the ZINB distribution [39, 55]. Alternatively, without requiring the definition of a parametric distribution, DISCERN disentangles the prediction of expression values and the dropout probability using two decoder heads. The first decoder head models the expression values and is trained using the Huber loss [119], defined as:

$$L_\delta(x, \hat{x}^{count}) = \frac{1}{d_x} \sum_{i=1}^{d_x} \begin{cases} \frac{1}{2}(x_i - \hat{x}_i^{count})^2 & \text{for } |x_i - \hat{x}_i^{count}| \leq \delta, \\ \delta\left(|x_i - \hat{x}_i^{count}| - \frac{1}{2}\delta\right), & \text{otherwise.} \end{cases} \tag{2.3}$$

where $x \in \mathbf{R}^{d_x}$ is the input expression vector for a cell, $\hat{x}^{count} \in \mathbf{R}^{d_x}$ the predicted expression vector for a cell from the first decoder head, d_x the number of genes, and δ a threshold deciding between the two conditions of the Huber loss. Considering mean-variance scaled data (see Section 2.3), this loss weights values far from the mean lower than values close to the mean. So, when applied to scRNA-seq data, the loss enables the model to learn a more robust expression estimate.

The second decoder head aims to estimate the dropout probabilities by comparing predicted dropout probabilities with the binary expression matrix. In comparison to the estimation of parametric distribution, e.g. in [120], has the advantage that the reconstruction of expression values and dropouts can be learned independently such that the model can learn batch-specific expression and dropout patterns. Furthermore, exact zeros can be retained, which can be important for the assumptions of downstream applications. This architectural choice of an independent second decoder head is novel for scRNA-seq analysis. For this second decoder head binary cross entropy is used to compute the loss values:

$$H(x^{dropout}, \hat{x}^{dropout}) = -\frac{1}{d_x}\sum_{i=1}^{d_x} x_i^{dropout}\log(\hat{x}_i^{dropout}) + (1 - x_i^{dropout})\log(1 - \hat{x}_i^{dropout}) \tag{2.4}$$

where $x^{dropout}$ is the expected binary expression matrix and $\hat{x}^{dropout}$ is the predicted binary expression matrix (probability of dropout). Combining the individual loss functions Equations (1.6), (1.7), (2.3) and (2.4) and adding the weighting factors λ_{prior}, λ_{sigma} and $\lambda_{dropout}$, the objective function can be formulated as:

$$L = L_\delta(x, \hat{x}^{count}(z)) + \lambda_{prior} \cdot \mathcal{D}_Z(q_z, p_z) + \lambda_{sigma} \cdot S_\sigma(x) + \lambda_{dropout} \cdot H\big(\mathcal{I}_{>0}(x), \hat{x}^{dropout}(z)\big) \tag{2.5}$$

Each layer of the neural network, except for the two decoder heads and the output layers of the encoder network, are followed by CLN using Equation (2.2) and the mean and variance estimate $\mu = \mu_i$ and $\sigma^2 = \sigma_i$ as defined in LN. To achieve non-linearities in the network after each CLN, the Mish (Equation (2.6)) [121] activation function is applied.

$$\mathrm{mish}(x) = x\tanh\big(\log(1 + \exp(x))\big) \tag{2.6}$$

Mish was shown to improve performance for deep neural networks in computer vision compared to often used activation functions, for example, Rectified linear units (ReLU) [121].

2.2.2 Regularization

Regularization is a common approach to prevent a machine-learning model from remembering the training data (overfitting) and helps the model to generalize to unseen examples [84]. While generalization plays an important role in classification and regression tasks [84], it is of minor concern when the model is only applied to training data. This is for example the case for data transformations and dimensionality reduction. However, even for dimensionality reduction generalization was shown to be beneficial [122].

Thus, several approaches for regularization were implemented in DISCERN. First, as discussed in Section 1.4.4, a regularization term is directly added to the loss function. Furthermore, after each activation function random removal of values (dropout) [123] is applied. This prevents, to some extend, co-adaptation of weights in the neural network [123]. Furthermore, activity regularization [124] is applied on the CLN. Regularization on network weights is widely applied to improve generalization [125, 126]. In the case of the W_β weight regularization is not possible, because if the number of conditions is unbalanced, the weights of the low abundant condition are regularized stronger than they are updated during training. This would lead to no or low learning of the weights for the low abundant conditions. Therefore, activity regularization is used, which enforces regularization depending on $W_\beta v^T$ and thus regularization is only applied if the condition is present in the current training step. Finally, early stopping was used to stop the training if the validation loss does not improve in further training epochs. Since the early-stopping criterion could be satisfied in the very first epochs of training, a minimum number of epochs is enforced. This delay was implemented to prevent too early stopping in the optimization procedure.

2.2.3 Training

DISCERN is trained on a combination of complete data sets with multiple batches. The weights of DISCERN are jointly optimized using the Rectified Adam optimizer [127]. Rectified Adam was developed to address the shortcomings of the widely used Adam optimizer [128]. Adam was shown to be successful in several optimization problems, however, can result in local optima, especially in the first epochs of training [127]. Thus often warmup, i.e. training using lower learning rates, in initial epochs is applied [129]. To overcome the problem of warmup parameter selection, Rectified Adam was developed [127], which is a version of Adam, where the variance of the learning rate, especially in the first epochs, is rectified [127]. Thus, no warmup is needed and careful adaption of the warmup parameters can be omitted.

2.2.4 Inference & Reconstruction

During model training two outputs, the estimated count matrix $\hat{x}^{count}$ and the estimated dropout rate $\hat{x}^{dropout}$ are used for loss calculation and model optimization. To acquire a count matrix $\hat{x}$, which can be used for downstream analysis, $\hat{x}^{count}$ is sampled using probabilities in $\hat{x}^{dropout}$ according to Equation (2.7).

$$\hat{x}_{i,j} = S(\hat{x}^{dropout}_{i,j}, \hat{x}^{count}_{i,j}) = \begin{cases} \hat{x}^{count}_{i,j} & \text{if } \hat{x}^{dropout}_i i, j \geq q \qquad \text{with} \quad q \rightsquigarrow \mathcal{U}_{[0,1]}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

for a cell i and a gene j with q sampled ($\rightsquigarrow$) from $\mathcal{U}_{[0,1]}$, which is a uniform distribution over the interval $[0, 1]$. If the estimated dropout rates $\hat{x}^{dropout}$ are a good estimate of the true dropout rate in the scRNA-seq data, using this sampling approach, the true distribution of scRNA-seq can be reconstructed. During model training, the binary cross-entropy (see Section 2.2.1) is used to enforce that this requirement is satisfied. This style transfer approach allows to use any downstream method, which was developed for the analysis of scRNA-seq data, because it was explicitly developed to retain the properties of scRNA-seq and thus statistical assumptions made by the downstream applications are likely to hold also for DISCERN reconstructed data. Furthermore, during inference, the random encoder is converted to a deterministic encoder, by not applying the reparameterization trick (Section 1.4.3) and using the estimate of the mean μ_i directly as embedding $z_i = \mu_i$ for decoding. This is commonly done for VAEs [88] and WAEs [104]. Thus the inference can be described as $\hat{x} = S(G(F_z(x, v), v))$ for the sampling process S, the decoder G, the encoder with deterministic output F_z and the conditioning variable v.

In the following, for the decoding and sampling process $\hat{x} = S(G(z, v))$ is denoted by $\hat{x} = G_S(z, v))$. It can be assumed that z_i is independent of the conditioning variable v_i by using a encoder F_z, which removes the effect of the condition from x_i by encoding it to z_i. The decoder G_S can then be applied to transform z_i with respect to any conditioning v' to $\hat{x}_{v'}$. This process will be called reconstruction (or projection) to a reference v'. It can be performed for any conditioning v', which was used during training in the encoder and decoder, without re-training the network. Reconstruction can be used to transform an scRNA-seq data set to a single batch label and with this, it is able to transfer the quality and style between multiple scRNA-seq batches. The complete workflow for reconstruction using DISCERN, including model training, is depicted in Figure 7.

2.2.5 Hyperparameters

Besides the learnable weights (often called parameters) of DISCERN, several non-learnable values, called hyperparameters, are required. These hyperparameters include architectural choices, for example, the reconstruction loss, activation function, the number of fully-connected layers, their size, and the number of embedding dimensions (Section 2.2.1). Other hyperparameters influencing the model training and its regularization include the choice of the optimizer, its parameters, the batch size, dropout rates of encoder and decoder, and the early stopping hyperparameters (Sections 2.2.2 and 2.2.3). Furthermore, the objective function contains three scaling factors λ_{prior}, λ_{sigma}, $\lambda_{dropout}$ (see Equation (2.5)) and a threshold value for the Huber loss (Equation (2.3)). An overview of the main hyperparameters and their default values can be found in Table 3.

Hyperparameter optimization

To find optimal values for hyperparameters of any deep-learning model, often hyperparameter optimization is performed [130]. Together with the DISCERN framework, a generic interface for hyperparameter optimization of DISCERN was implemented. This generic framework has the advantage of allowing the use of multiple method for hyperparameter optimization and can easily be extended for future algorithms. To assess DISCERNs performance, the framework uses the classification performance of a Random Forest classifier between real cells and cells, which are called auto-encoded because they were encoded and decoded by the network using their original condition labels. The classification performance is summarized using the area under the receiver operating characteristic curve (AUROC). The hyperparameter selection of the most important hyperparameters was performed using

Table 3: Overview of most important hyperparameters and their default values for DISCERN.

Hyperparameter	Default values
Architectural hyperparameter	
Activation function	Mish
Number of fully-conncected layers	3
Encoder layer size	1 024-512-256
Decoder layer size	256-512-1 024
Embedding size n_{embed}	48
Learning rate	1×10^{-3}
Training hyperparameter	
Optimizer	Recified Adam
Recified adam decay rates	0.85; 0.95
Batch size	192
Encoder dropout rate	0.4
Decoder dropout rate	0
Early stopping – patience	30 epochs
Early stopping – delta	0.01
Early stopping – delay	5 epochs
Loss hyperparameter	
Reconstruction loss	Huber
Huber loss threshold δ	9
λ_{sigma}	1×10^{-8}
$\lambda_{dropout}$	1×10^{5}
λ_{prior}	1 500

grid search in Chapter 3.

2.2.6 Implementation

DISCERN is implemented using the TensorFlow framework (v2.1.0). This enables fast, GPU-accelerated, deep-learning model training. All hyperparameters can be specified using a JavaScript Object Notation (JSON) configuration file. All parts of the implementation are tested using the unit-test framework of pytest.

To compute the MMD-penalty, three implementations are provided: A Tensorflow based version of GPU-accelerated computations used for model training, an implementation in Python and an implementation in C. The latter two were used for manual evaluation and during hyperparameter optimization. The C-based implementation has a Cython-interface [131] to access it in Python. The hyperparameter optimization is implemented using the ray[tune] library [132] and supports their techniques for fast hyperparameter optimization. The hyperparameter search space can be defined using a JSON file.

Running time and memory usage The running time of deep learning models using fully connected layers, i.e. DISCERN, is linear in the number of cells and the number of training epochs during training [133]. The early stopping mechanism, however, can in practice considerably reduce the running time and improve model performance. The running time for each epoch is additionally dependent on the size of the mini-batches. Since DISCERN needs to keep the expression matrix in memory, the memory requirement is linear in the number of cells and genes for training and inference.

Code availability DISCERN is available at the Python Package Index (PyPI) `https://pypi.org/project/discern-reconstruction/` for users. The source code can be found at `https://github.com/imsb-uke/discern`. The exact versions of packages used for all analyses are provided using Poetry (`https://python-poetry.org/`).

2.3 Data preprocessing

For combining multiple batches b_k, generated for example by different sequencing technologies, the following approach was used to combine the respective raw expression datasets $M^{b_k} \in \mathbf{R}_{\geq 0}^{n_k \times m}$ including n_k cells and m genes.

To obtain a such dataset (block matrix) $M = \begin{bmatrix} M^{b_1} \\ M^{b_2} \\ \dots \end{bmatrix} \in \mathbf{R}_{\geq 0}^{n \times m}$, where $n = \sum_k n_k$ the intersection of detected genes was taken such that $\sum_i^{n_k} M_{i,j}^{b_k} > 0$ for all batches b^i and all genes $j \in m$. This ensures that all genes m are expressed in at least one cell in all batches. Different approaches, e.g. the union of genes noted as $\sum_i^{n_k} M_{i,j}^{b_k} = 0$ for some batches b^i and genes $j \in m$, would require filling in missing values with zeros, which would introduce additional noise and batch effects. These additional batch effects could potentially be very different from other batch effects in the data since they only affect some genes, but all cells in one or multiple batches. This would increase the complexity of the imputation problem.

The preprocessing of counts was performed using recommended steps as proposed by [25]. Genes expressed in less than three cells and cells expressing less than 10 genes were considered insufficiently sequenced and thus they were removed. The total number of remaining counts per cell was normalized to a value of $20\,000$ (called library size normalization). This normalizes the total amount of counts (RNA molecules) in each cell to a fixed value, assuming that all cells contain an equal amount of RNA and differences are only due to sampling [66].

Subsequently, the count data were log-transformed to adjust the distribution of the expression data closer to a Gaussian distribution and to alleviate the mean-variance dependency of expression data [66]. Further scaling is not recommended for downstream applications in scRNA-seq data analysis [66], but can improve the performance of deep neural networks [134]. Fortunately, mean-variance scaling can be reversed on the reconstructed expression data, such that downstream applications can be applied on the unscaled data. This will be discussed below.

Therefore, the expression vector $x_{.,j}$ for the j-th gene is scaled to a mean of zero and a variance to one based on its mean $\bar{x}_j$ and standard deviation s_j as follows: $\mathrm{scale}(x_{.,j}) = \dfrac{x_{.,j} - \bar{x}_j}{s_j}$.

For downstream applications, after the application of DISCERN, the expression data can be rescaled to obtain a distribution with the original mean and variance as follows:

$\mathrm{rescale}(x_{\cdot,j}) = \hat{x}_{\cdot,j} s_j + \bar{x}_j.$

The unscaled data can then be used for downstream applications as recommended by Luecken & Theis [66].

An overview of the DISCERN preprocessing pipeline and its use in DISCERN reconstruction can be found in Figure 8.

2.4 Data sets

To evaluate the quality of expression reconstruction methods, several publicly available data sets were used [5, 45, 135]. The evaluation and benchmarking of DISCERN is based on the data sets described in Section 2.4.1–Section 2.4.7. A tabular summary of all data sets is available in Appendix A and Table 10. In the following, sequencing technologies will be denoted using capitalized names, whereas batches sequenced with corresponding sequencing technology will be indicated using lowercase letters. If necessary, the suffixes "-lq" and "-hq" will be added to denote low quality or high quality, respectively.

2.4.1 Pancreas

The pancreas data set consists of different scRNA-seq experiments profiling diabetes-related changes in the pancreas [135]. The collection was furthermore widely used for batch correction benchmarks due to its extensive number of cell types and sequencing technologies [49] allowing for a comprehensive evaluation of sequencing technology-related batch effects. The data set was sequenced using five sequencing technologies (Smart-Seq2, Fluidigm C1, CelSeq, CEL-Seq2, inDrop) comprising 13 cell types (alpha, beta, ductal, acinar, delta, gamma, activated_stellate, endothelial, quiescent_stellate, macrophage, mast, epsilon, Schwann). In total, the data set contains $14\,890$ cells. The data set, including cell type and batch annotation, is available as panc8.SeuratData (v3.0.2) [135]. The Smart-Seq2 batch was selected as high-quality (smartseq-hq), due to its high number of expressed genes and reasonable number of cells (Table 11).

2.4.2 Difftec

The difftec data set is a collection designed for a systematic comparative analysis of scRNA-seq methods [5]. It was sequenced using seven partially related sequencing technologies (10x Chromium v2, 10x Chromium v3, Smart-Seq2, Seq-Well, inDrop, Drop-seq, CEL-Seq2), each with at least two technical replicates. In total 31 021 cells are present in 10 different cell types (Cytotoxic T cell, CD4$^+$ T cell, CD14$^+$ monocyte, B cell, Natural killer cell, Megakaryocyte, CD16$^+$ monocyte, Dendritic cell, Plasmacytoid dendritic cell, Unassigned).

The data set is available as pbmcsca.SeuratData (v3.0.0) [5]. The chromium-v3 batch, sequenced by 10x Chromium v3, was selected as hq because 10x Chromium v3 was suggested to be more sensitive in detecting low expressed genes [5] and the batch has a reasonable number of detected genes (Table 11).

2.4.3 snRNA-seq & scRNA-seq

This data set was created for the evaluation of a single-cell and single-nuclei analysis toolbox [45]. For the evaluation of reconstruction methods, only the liver biopsy sample (HTAPP-963) of metastatic breast cancer was considered, because this biopsy was sequenced using snRNA-seq and scRNA-seq.

The data set contains a total of 12 423 cells and eight cell types (Epithelial cells, Macrophages, Hepatocytes, T cells, Endothelial cells, Fibroblasts, B cells, NK cells) sequenced using the Chromium v3 technology on an Illumina HiSeq X sequencer. Data collected using snRNA-seq and scRNA-seq data differ in the cellular composition, the availability of other modalities (e.g. CITE-seq), and in the amount of detected gene expression [5, 45].

From a biological perspective the quality of snRNA-seq and scRNA-seq cannot be evaluated very well because they provide different insights into cellular expression. While snRNA-seq only allows sequencing of the transcript from the nucleus, scRNA-seq is able to sequence transcripts of the cytoplasm and the nucleus simultaneously. Thus, the measured gene expression differs between snRNA-seq and scRNA-seq. Here, the scRNA-seq batch (sc) was considered as hq due to its high number of counts per cell (Table 11).

2.4.4 citeseq

The citeseq batch is a 10x Chromium-based sequencing data set of healthy human PBMCs for 6 cell types (B cells, CD4 T cells, NK cells, CD14$^+$ Monocytes, FCGR3A$^+$ Monocytes, CD8 T cells) [136]. In addition to the expression data, the abundance of 13 surface proteins was measured using CITE-seq [137]. This offers ground-truth information on the cell type marker genes and serves as a proxy for the expression of the corresponding genes. The version of the 10x Chromium Kit was not provided by the authors. Cell-type annotations were used as provided by [55]. The citeseq batch was used in combination with the bulk-hq batch (Section 2.4.5).

2.4.5 bulk

Ota *et al.* provides a large data set of Fluorescence Activated Cell Sorting (FACS)-sorted and bulk-sequenced immune cell types from 416 donors, out of which 79 are healthy. It consists of the 9 852 samples and corresponding to 28 sorted immune cell types (Naive CD4, Memory CD4, TH1, TH2, TH17, Tfh, Fr. I nTreg, Fr. II eTreg, Fr. III T, Naive CD8, Memory CD8, CM CD8, EM CD8, TEMRA CD8, NK, Naive B, USM B, SM B, Plasmablast, DN B, CL Monocytes, Int Monocytes, NC Monocytes, mDC, pDC, Neutrophils, LDG) with $> 99\,\%$ purity [138].

The sequencing was done using the SMART-seq v4 Ultra Low Input RNA protocol (Takara Bio). Due to the low dropout rate compared to scRNA-seq data, this data set was always considered of hq.

2.4.6 kidney-lq (snRNA-seq) & kidney-hq (scRNA-seq)

The kidney data set was sequenced using scRNA-seq and snRNA-seq of 9 patients with acute kidney injury. The sequencing was performed using 10X Genomics Chromium technology v3 providing scRNA-seq data for 82 701 cells in total where the data for 52 934 cells comes from snRNA-seq and the data for 29 767 cells comes from scRNA-seq. No cell type annotation is provided with this data set, but it is part of a bigger data set about acute kidney injury [139]. Similar to Section 2.4.3, the scRNA-seq was used as an hq batch.

2.4.7 covid-lung & covid-blood

The COVID-19 data set consists of blood and bronchoalveolar lavage (BAL) samples from three patients with bacterial pneumonia (Bac17B, Bac18B, Bac19B) and seven patients with SARS-CoV-2 infection (Sar01B, Sar02B, Sar03B, Sar04B, Sar05B, Sar07B, Sar08B) [140]. In total 155 706 cells were sequenced using the 10X Chromium Single Cell V(D)J Reagent Kit v1.1. In addition, TCR-seq was performed to get information on the T cell receptor sequence. During development, each T cell creates a unique T cell receptor (TCR) sequence, and thus clonally expanded T cells can be identified using TCR-seq. The lung data was investigated in depth, and the analysis of the blood data was limited to cell type identification in the original publication [140]. Cell type annotation for the BAL was obtained from [140].

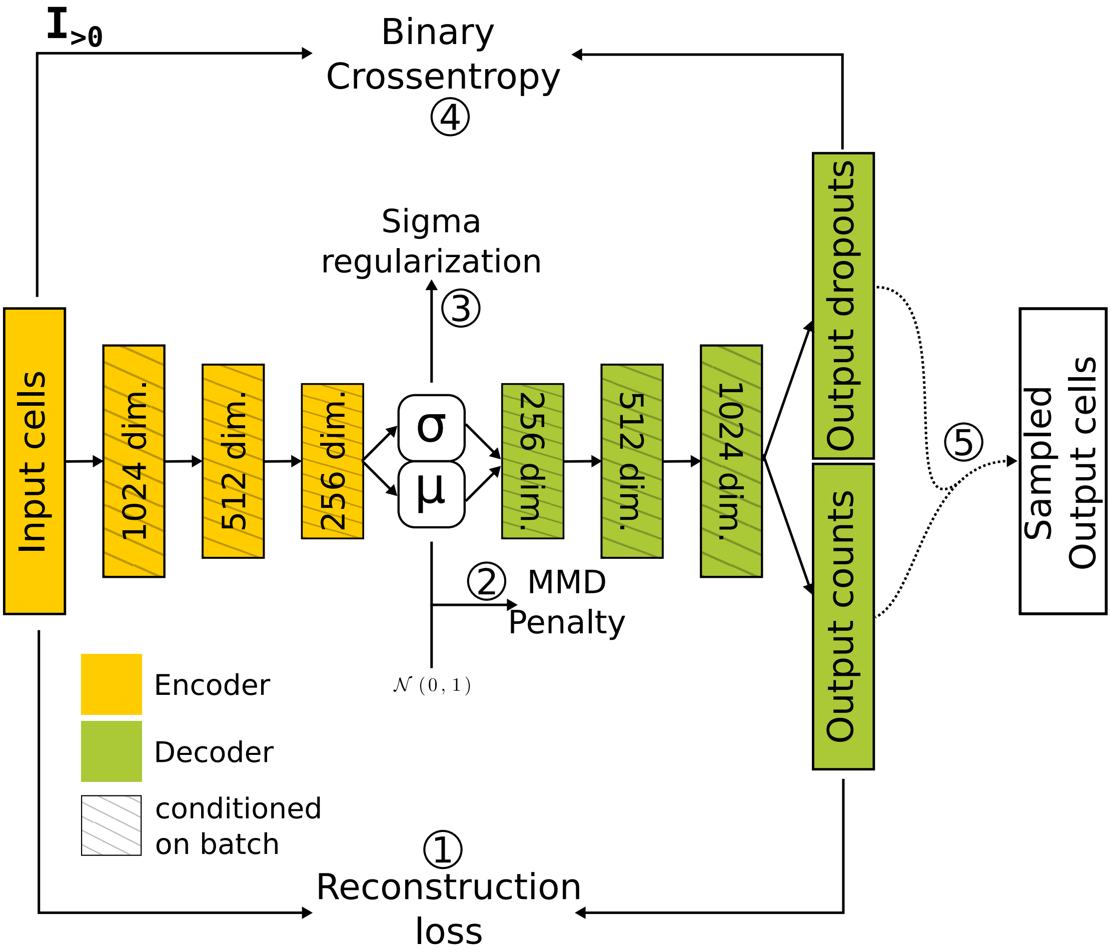


Figure 6: Overview of the DISCERN neural network architecture. DISCERN consists of a random encoder (yellow) and a deterministic decoder (green). Encoder and decoder can be conditioned on the batch information (see Section 2.1). DISCERN aims to optimize an objective function combining a (1) reconstruction loss (Equation (2.3)), (2) a MMD-penalty term for comparing prior and aggregated posterior distribution (Equation (1.6)), (3) a sigma regulation term to prevent the random encoder from collapsing to a deterministic one (Equation (1.7)), and (4) a binary cross-entropy term for learning the probability of a dropout event (Equation (2.4)). (5) A count matrix is obtained during inference by sampling from the estimated counts with the estimated dropout probabilities using Equation (2.7).

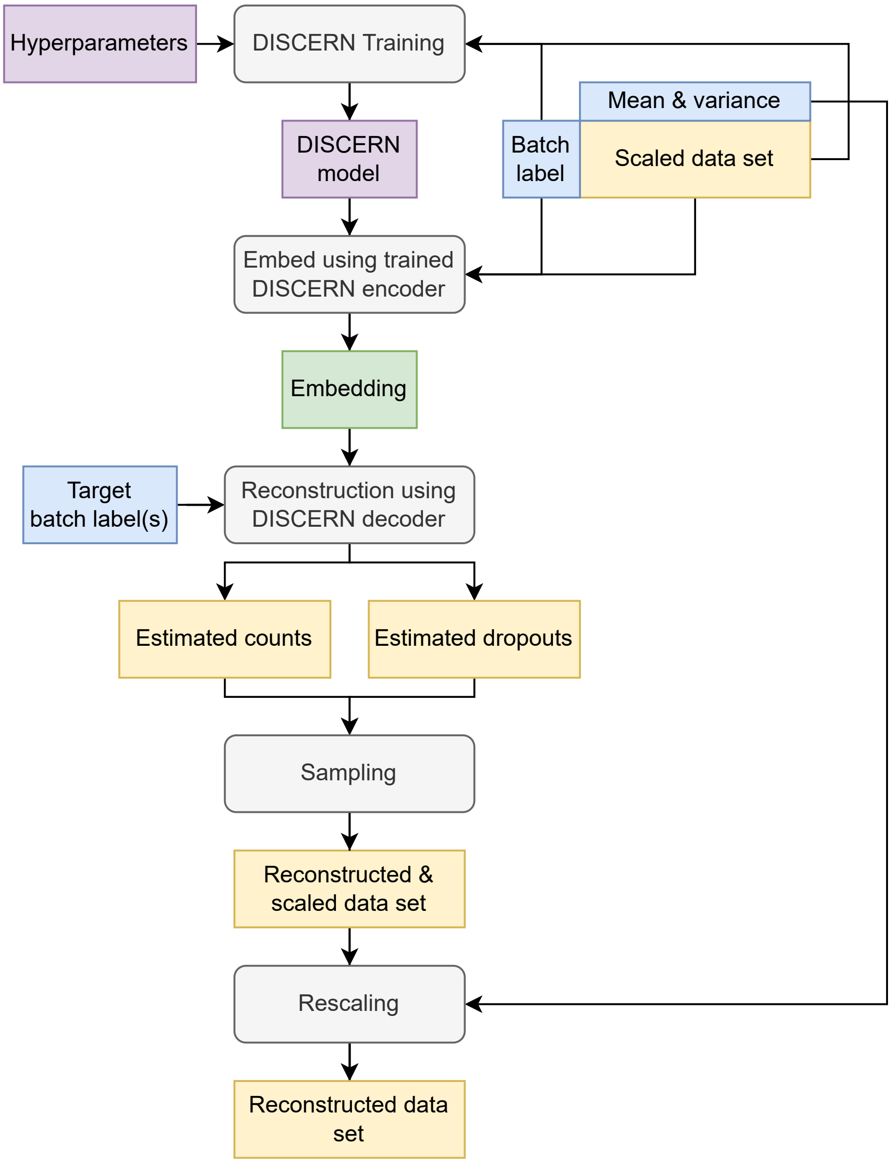


Figure 7: Overview of the DISCERN-based reconstruction procedure. Data sets and expression matrices are marked by yellow rectangles. Additional metadata (e.g. gene-wise mean and variance and cell-wise batch labels) is depicted by blue rectangles. For DISCERN hyperparameters and model files violet is used. The data set embedding is shown in green and applied functions as gray boxes. The input (scaled data set) is provided by the DISCERN preprocessing pipeline (Figure 8) and the hyperparameters are specified using a JSON template. The target batch labels are provided by the user.

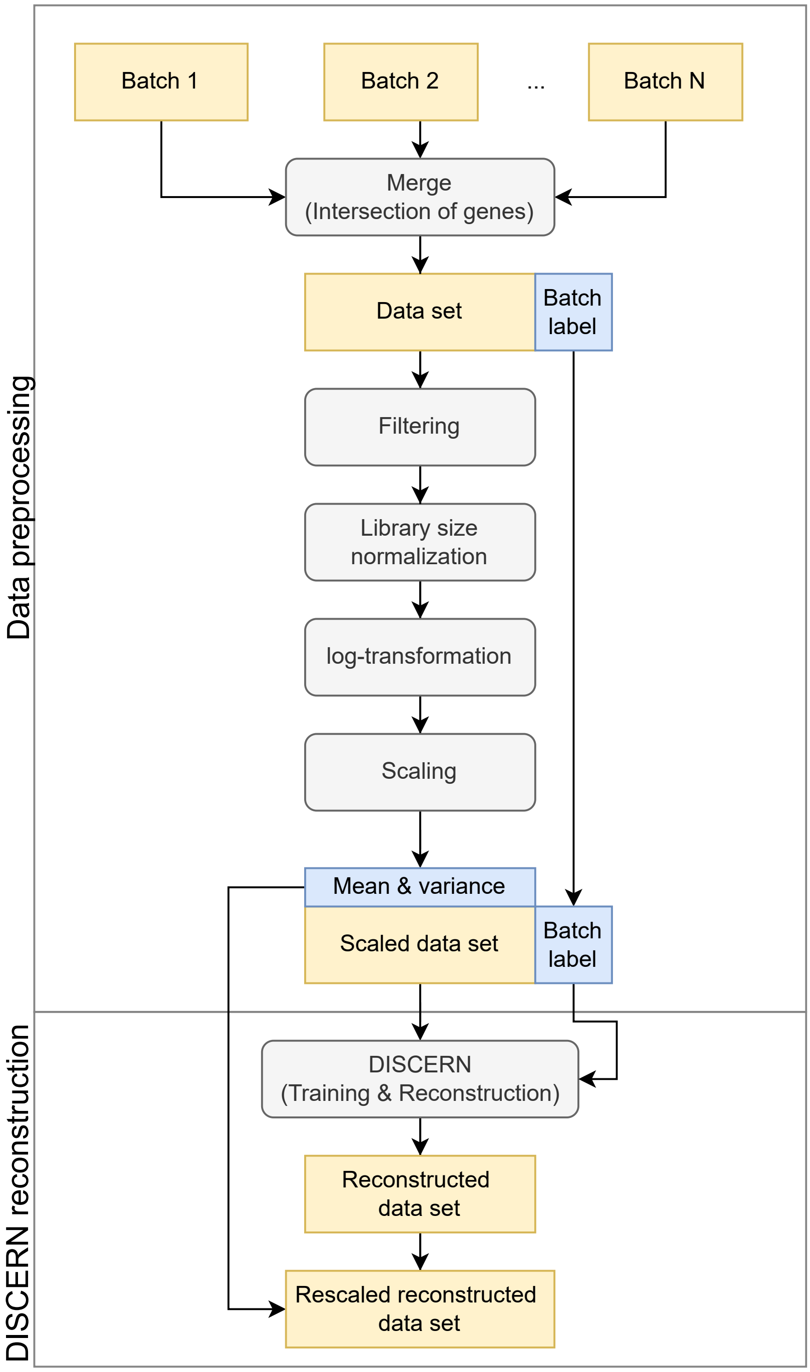


Figure 8: Overview of DISCERN preprocessing and application. Data sets and batches are marked using yellow rectangles, additional metadata (e.g. gene-wise mean and variance and cell-wise batch labels) with blue rectangles, and applied functions with gray boxes. The upper part belongs to the preprocessing scheme implemented in the DISCERN pipeline and the lower part depicts the reconstruction procedure using DISCERN and its inputs from the preprocessing pipeline.

3 Results

The reconstruction of absent gene expression values is still an open problem in scRNA-seq, as outlined in Section 1.1. Batch effects can introduce further unwanted variations in the analysis. However, the difference in quality lends itself to simultaneously impute and batch correct the data, called expression reconstruction. DISCERN was designed to realistically reconstruct gene expression in scRNA-seq data with a related hq dataset. This process requires batch correction and imputation of technically absent values [34]. The complete workflow requires data preprocessing (Section 2.3), model training (Section 2.2.3), and reconstruction (Section 2.2.4) and is outlined in graphical overview Figures 7 and 8.

For reliable modeling using deep learning methods, such as DISCERN, the choice of hyperparameter is important. Poorly tuned hyperparameters result in a loss of performance or failed model training [130]. The effect of the hyperparameters on DISCERN was evaluated by comparing the mean expression of the high-quality batch smartseq-hq from the pancreas data set to the mean expression of all other batches after reconstruction with the smartseq-hq as a reference. The pancreas data set was selected for this purpose because it is the most comprehensive dataset in terms of cell number and number of sequencing technologies. Pearson correlation between the mean expression of the reconstructed data and the smartseq-hq batch was used as a measure of model performance. The hyperparameters λ_{prior}, λ_{sigma}, $\lambda_{dropout}$ and the number of dimensions for the embedding (n_{embed}) were evaluated using three different values for each of these. In all $3^4 = 81$ combinations, DISCERN achieved a Pearson correlation of 0.949 to 0.965 (mean 0.957, Figure 9). Thus, DISCERN is robust with respect to the choice of hyperparameters in the pancreas data set. The best performance (0.965) was achieved using $n_{embed} = 56$, $\lambda_{prior} = 2\,000$, $\lambda_{sigma} = 1 \times 10^{-9}$ and $\lambda_{dropout} = 5\,000$, whereas the worst performance (0.949) was obtained with $n_{embed} = 10$, $\lambda_{prior} = 1\,500$, $\lambda_{sigma} = 1 \times 10^{-7}$ and $\lambda_{dropout} = 15\,000$.

For the following experiments, the parameters were conservatively selected by using a combination, which achieves close to average performance on the pancreas. These parameters are $n_{embed} = 48$, $\lambda_{prior} = 1\,500$, $\lambda_{sigma} = 1 \times 10^{-8}$, $\lambda_{dropout} = 10\,000$ and DISCERN achieves a correlation of 0.958 with these default parameters.

Further hyperparameter optimization such as data set-specific optimization for any other data set was not performed for DISCERN, because not tuning hyperparameters reflects a more realistic application of preprocessing tools in real analyses. For all experiments, the default hyperparameter of DISCERN was used and the grid search (Figure 9) was only used to assess the stability of performance using the default hyperparameter. In all benchmarks, competing methods were used with the default, author-provided hyperparameters.

3.1 Benchmarking expression reconstruction methods

In this section, the performance of DISCERN in comparison to other batch correction and imputation algorithms is assessed. Three use cases were evaluated: Batch correction, imputation using real data and *in silico* modified data. Not every tool was developed for all three use cases, except DISCERN, which is why a wide range of tools were selected to provide a fair comparison. scGEN [94], Seurat [53], trVAE [105] and scVI [55], are state-of-the-art batch correction tools. DCA [39], MAGIC [40], scImpute [41], DeepImpute [42], and CarDEC [43] are state-of-the-art imputation tools. These tools were used in the benchmarks.

All tools provide the possibility to generate a reconstructed expression matrix and are evaluated using classical scRNA-seq analysis methods. Seurat, DCA, MAGIC, scImpute, CarDEC, and DeepImpute do not provide the option to select a reference batch and are therefore only evaluated on the batch-independent reconstructed expression data.

3.1.1 Batch correction

Batch correction methods are evaluated by comparing their ability to mix the different batches to form homogeneous representations (batch mixing). Another important aspect is the preservation of biology, i.e. the cellular identity or cell types.

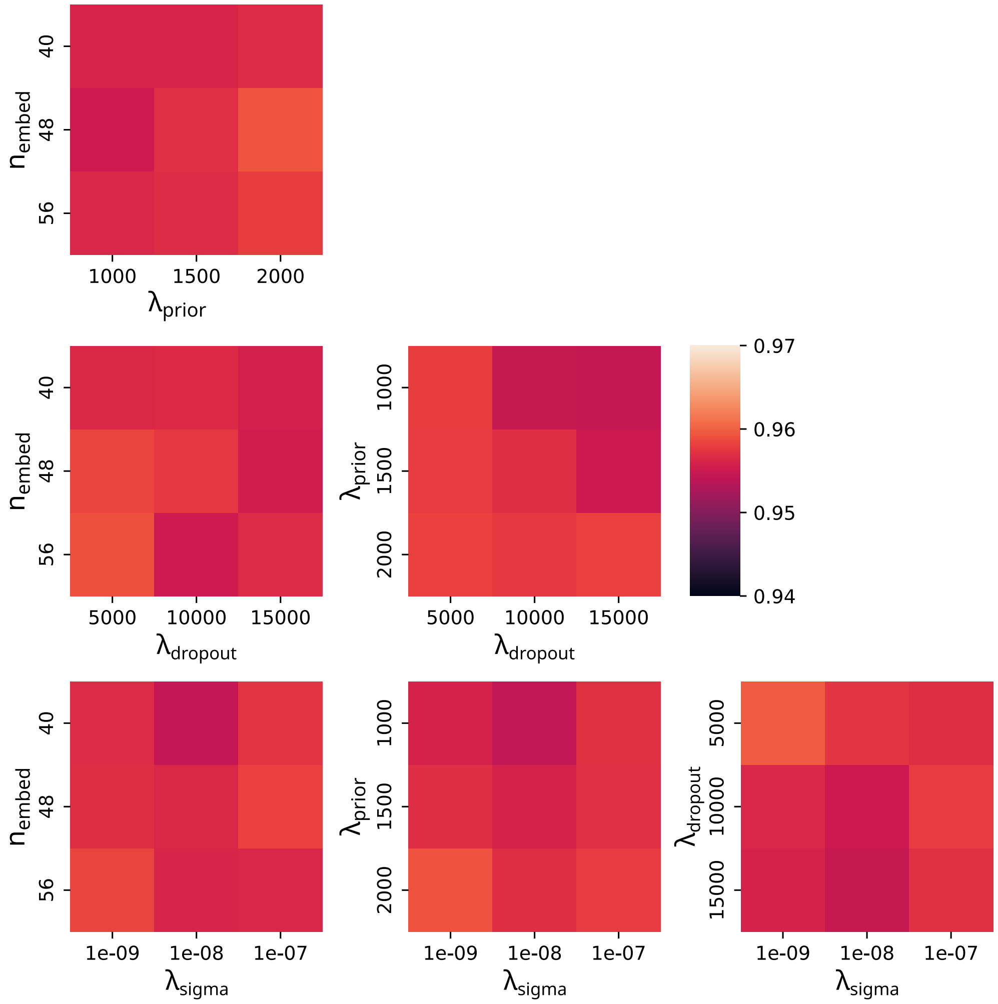


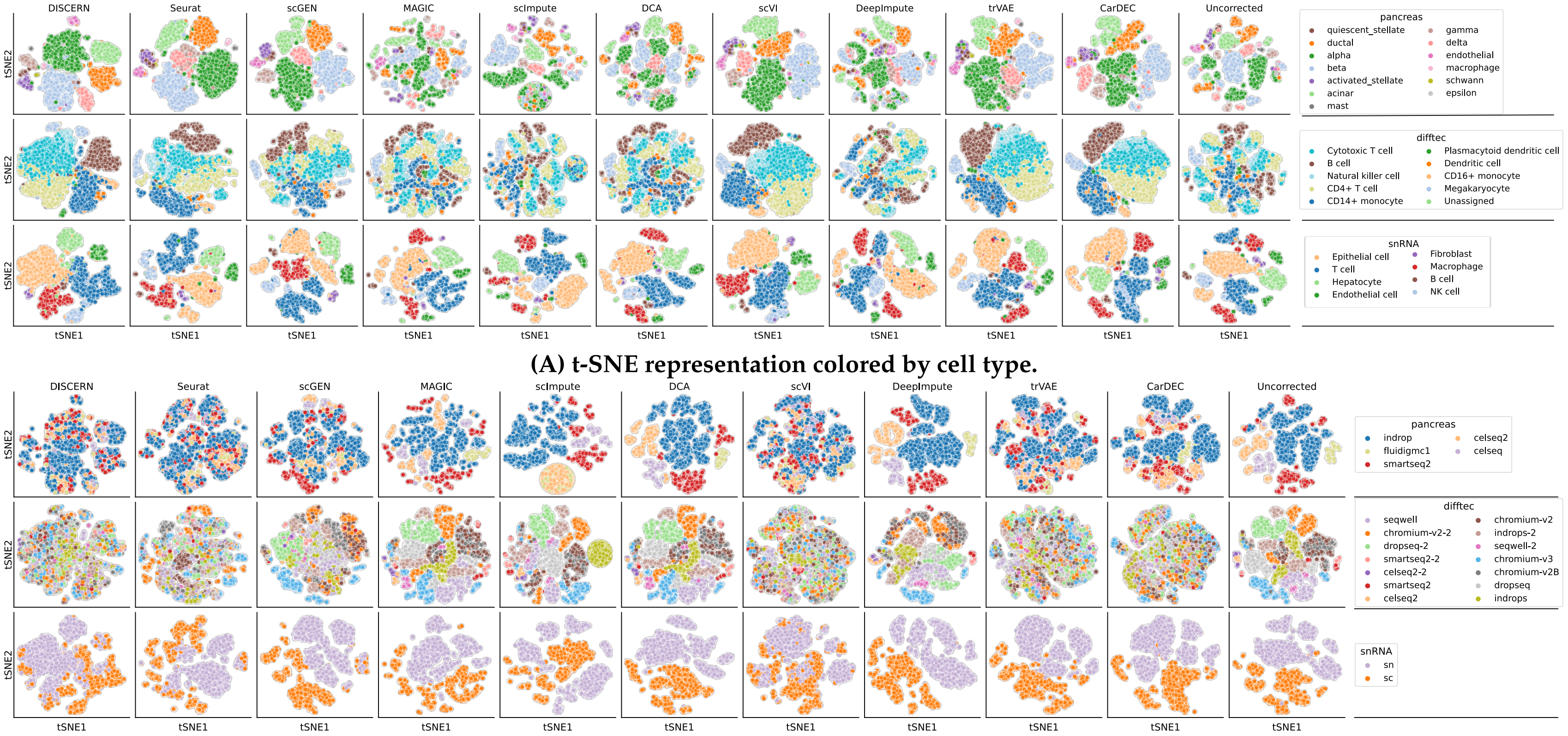
Figure 9: Average gene expression correlation in the pancreas dataset for multiple combinations of the four main hyperparameters of DISCERN. Pearson correlation was computed on the average gene expression per cell type between the uncorrected smartseq2-hq batch and all other batches reconstructed to the smartseq2-hq batch. All possible combinations between the four hyperparameters were trained. Each box in the heatmap represents the average of all models sharing this combination of hyperparameters. Embedding dimensions (n_{embed}) specifies the size the embedding provided by DISCERN, λ_{prior} is a scaling factor for the MMD penalty on the embedding dimensions and the prior, λ_{sigma} is a scaling factor of the loss on the estimated variance in the embedding dimensions and the $\lambda_{dropout}$ is the scaling factor of the cross entropy loss of the dropout estimation.

A very good batch-mixing performance can be achieved by randomly distributing cells in the embedding, which would, however, remove every biological insight.

Thus, batch mixing and biology preservation are contradicting [49]. However, appropriate batch mixing with preserved biology can improve marker gene discovery and cell type detection [106]. Batch correction performance was qualitatively assessed using the t-SNE representation and quantitatively evaluated using AMI, ARI, and the Silhouette score (Section 1.3.2). Batch correction evaluation was performed on the difftec, the pancreas, and the snRNA-seq data sets. As discussed in Section 2.4, the pancreas and the difftec data set cover a wide range of sequencing technologies and are widely used for batch effect correction evaluation. The snRNA-seq data provides the unique opportunity to test the capability of the methods to correct for differences in the isolation of the RNA. Usually, in scRNA-seq experiments clustering and dimensionality reduction are performed using the top principal components from PCA [66] of the scaled expression data after reconstruction. Therefore, these steps were applied for evaluation as well.

Qualitatively, except for the two imputation methods DeepImpute, and scImpute all methods can separate cell types in all three data sets (Figure 10A). Both methods especially perform bad on data sets with small batch effects, i.e. the pancreas and the difftec data sets. This could be because imputation methods are often only applied to data sets consisting of one batch and batch effects can overshadow cell type differences for the inference of the imputation parameter. Furthermore, except for CarDEC, no imputation method is able to remove the batch effect on all three data sets (Figure 10B).

However, on the pancreas data set, scGEN and CarDEC fail to integrate the smart-seq2 batch, whereas the other batch correction methods, i.e. DISCERN, Seurat, scVI, and trVAE can correct the batch effects. On the difftec data set only scGEN fails to integrate all batches, and keeps especially the chromium-v3 batch separate. The snRNA-seq data set shows the strongest batch effect and only DISCERN, Seurat and scVI are partly able to integrate the sc and the sn batch.



(A) t-SNE representation colored by cell type.

(B) t-SNE representation colored by batch.

Figure 10: t-SNE representation of the pancreas data st (first row), the difftec data set (second row), and the snRNA-seq data set (third row) after reconstruction. For the pancreas data set the smartseq2 batch, for the difftec data set the chromium-v3 batch, and for the snRNA-seq data set the sc batch was used as reference. 25 principal components were used for the t-SNE computation.

The qualitative analysis was performed using 25 principal components, however, the normal workflow of scRNA-seq data analysis involves a selection step of the top principal components [66]. To reflect this selection process, a varying number of principal components was used to compute the batch correction metrics. The influence of the number of principal components on the t-SNE representation can be seen in Figure 11. A higher number of principal components leads to a higher number of small clusters, whereas a low number of principal components leads to fewer clusters with more mixing between cell types and batches.

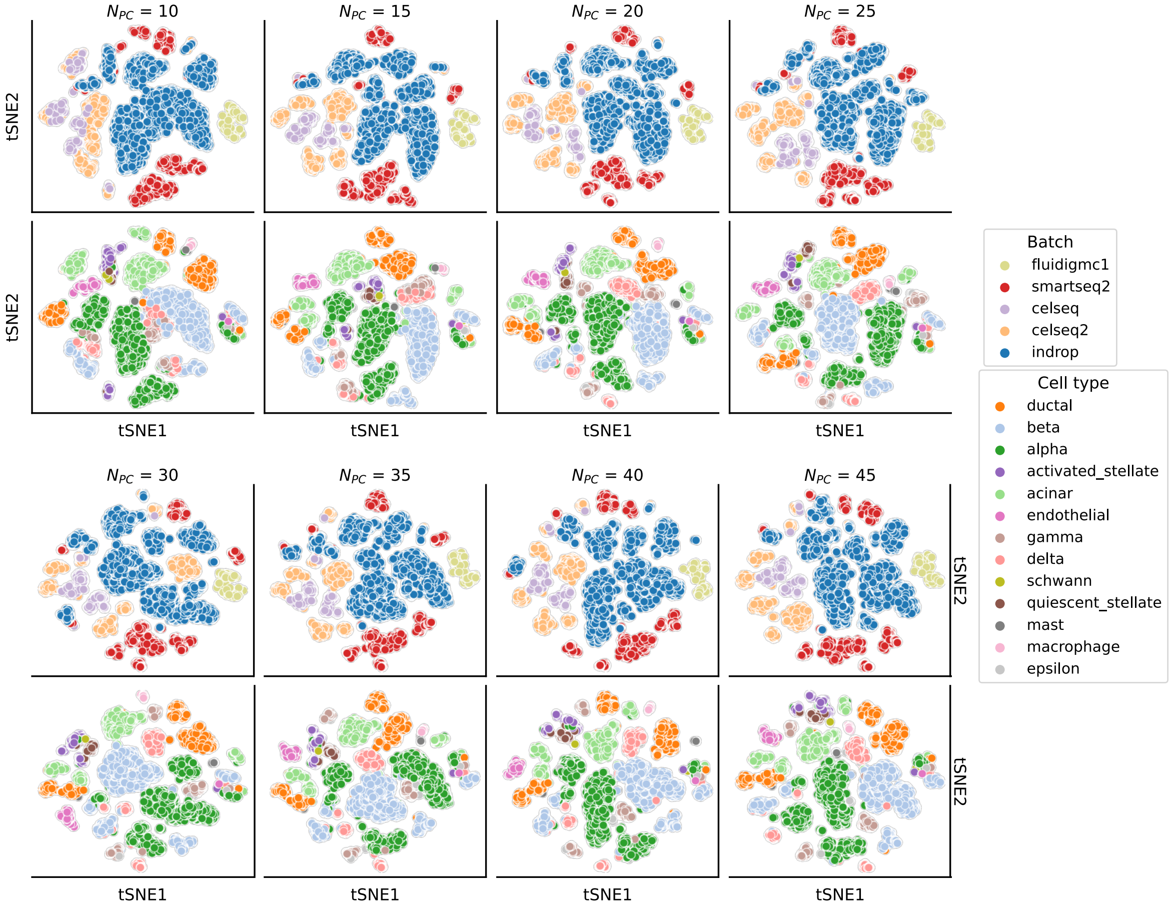


Figure 11: t-SNE representation of the pancreas data set without reconstruction for different numbers of principal components. The figure is split into two parts to improve readability. For the second part the location of the tSNE2-axis is swapped. The first row of both parts is colored by batch and the second row of both parts is colored by cell type. For low number of principal components (e.g. 10) a some cell types are overlapping, whereas for a high number of principal components (e.g. 45) several small clusters can be seen.

The Silhouette score directly evaluates the distances in the PCA embedding used for further downstream applications. DISCERN achieves the best performance for

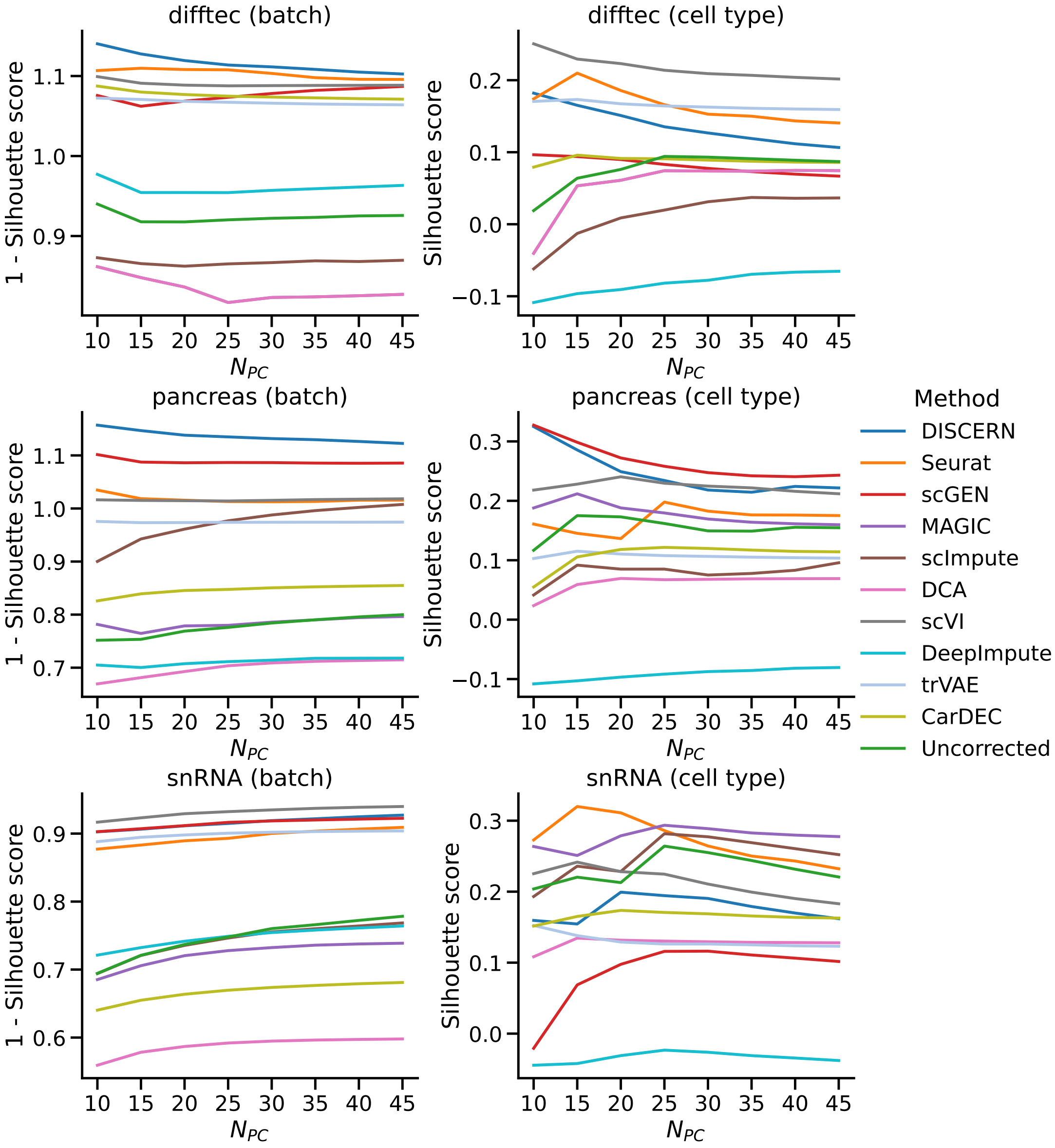


Figure 12: Silhouette scores for batch and cell type clusters on the difftec, the pancreas, and the snRNA-seq datasets for varying numbers of principal components (N_{PC}). The first column measures batch clustering ($1-$Silhouette score) and the second column cell type clustering. The larger the numbers the better the performance.

batch mixing on the pancreas and the difftec data set. On the snRNA-seq data set, DISCERN shares rank two with scGEN. scVI reached best batch mixing on the snRNA-seq data set (Figure 12, first column). Considering the cell type clustering (preservation of biology) no method consistently outperforms the others (Figure 12, second column). On the difftec data set scVI performs best, on the pancreas data set scGEN achieves the highest rank, and on the scRNA-seq data set Seurat and MAGIC perform best depending on the number of principal components used, e.g. for low number of principal components Seurat achieves better performance, whereas for high number of principal components MAGIC is the best performing method. DISCERN achieves ranks between two and six and thus similar performance as other batch correction tools.

The other two measures, ARI and AMI, are computed on clustering results. Therefore, Leiden clustering [141] was performed on the respective number of principal components N_{PC} using 20 different values for the resolution parameter. Leiden clustering is a commonly used clustering algorithm, which is implemented in scRNA-seq analysis tools kits, e.g. Seurat [53] and scanpy [142].

The resolution parameter can be interpreted as a threshold indirectly controlling the number of clusters by restricting the cluster density. For a value of N_{PC}, the best resolution was selected for batch mixing and cell type clustering.

DISCERN and scVI show very similar performance for batch mixing evaluated by ARI on the difftec and the snRNA-seq data set. On the pancreas data set DISCERN and Seurat show similar scores (Figure 13, first column), with scVI in the third rank. Considering cell type clustering on all three data sets DISCERN and scVI achieve similar performance (Figure 13, second column).

Using AMI for evaluating batch mixing and cell type clustering, the deep learning methods CarDEC, DISCERN, Seurat, and scVI perform best (Figure 14). DISCERN, CarDEC and scVI achieve similar performance in batch mixing and cell type clustering on the difftec data set. However, on the pancreas data set the performance of CarDEC drops drastically, and DISCERN, scVI, and Seurat perform best. Furthermore, scVI and DISCERN are the only methods showing good performance on the scRNA-seq data set (Figure 14).

These results can be summarized by the computation of the $\log_2$-fold-change improvement compared to uncorrected data (Figure 15). None of the tested methods consistently outperforms all other methods when considering all data sets. Addi-

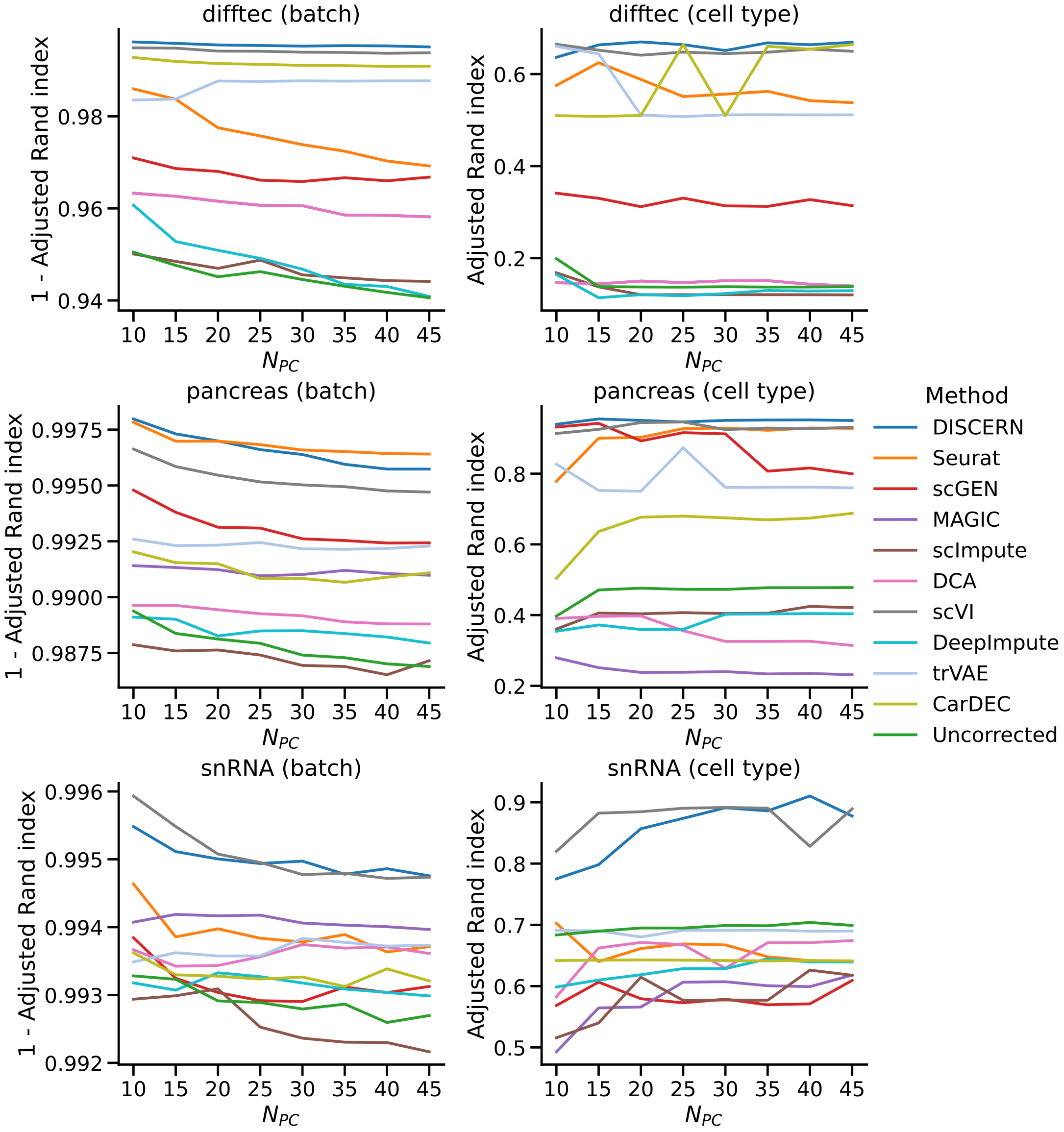


Figure 13: ARI for batch and cell type clusters on the difftec, the pancreas, and the snRNA-seq dataset for varying numbers of principal components. The first column measures batch clustering (1 - ARI) and the second column measures cell type clustering. Clustering was performed using the Leiden algorithm on 20 different resolutions per number of components. The best ARI for each number of components is displayed. Higher numbers indicate better performance.

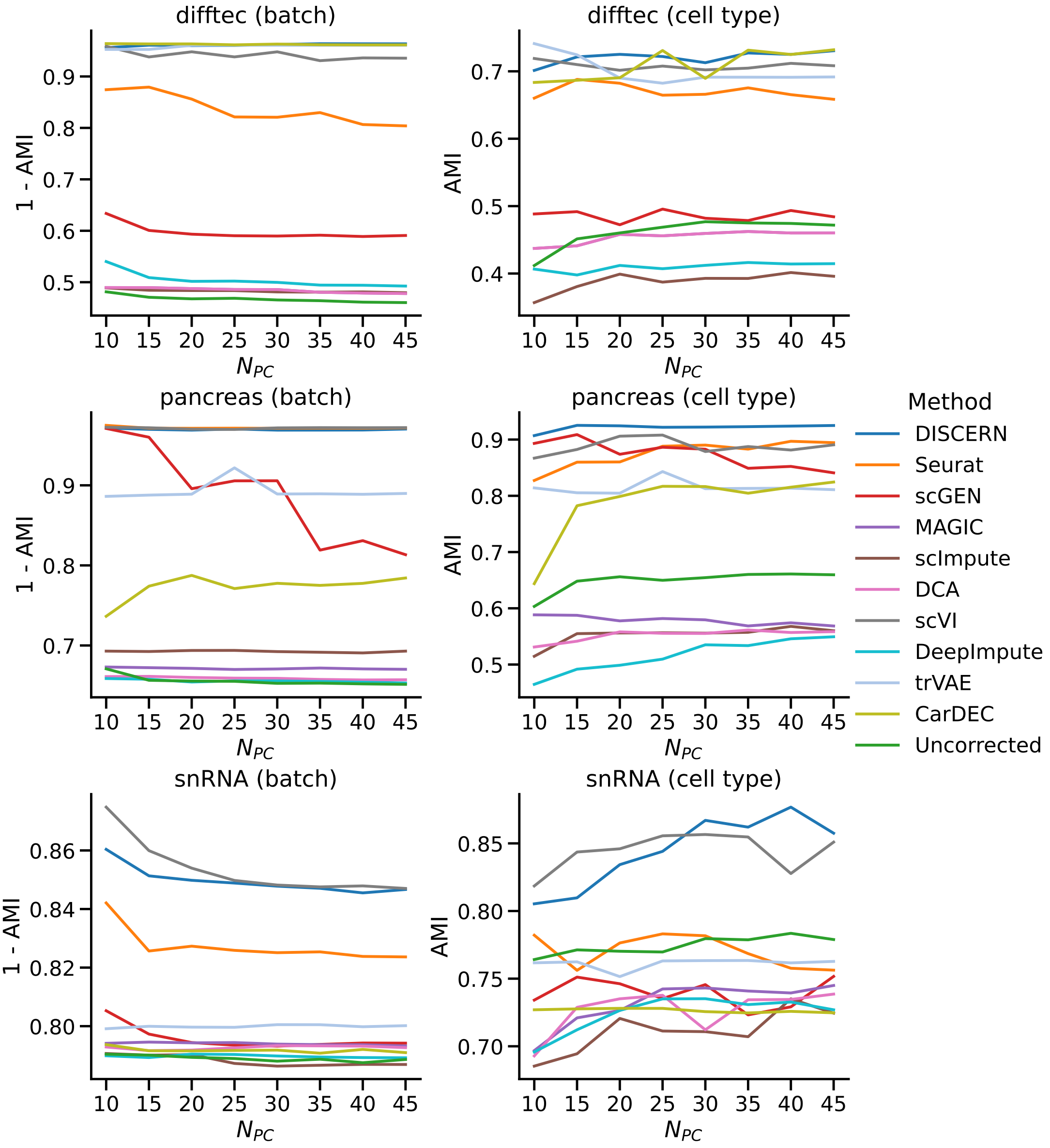


Figure 14: AMI for batch and cell type clusters on the difftec, the pancreas, and the snRNA-seq dataset for varying numbers of principal components. The first column measures batch clustering (1 - AMI) and the second column measures cell type clustering. Clustering was performed using the Leiden algorithm on 20 **different resolutions per number of components. The best AMI for each number of components is displayed. The larger the numbers the better the performance.**

tionally, the three tested metrics do not always lead to the same ranking.

Since they all evaluate different aspects this is not unexpected but makes interpretation of the results difficult. While Silhouette scores evaluate the PCA representation, ARI and AMI represent clustering results. The ARI was found to be a better measure if there are large equal-sized clusters whereas the AMI gives more reliable results for unequal-sized clusters [143]. Both are valid scenarios for scRNA-seq analysis. Therefore, all three metrics were considered for analysis (Figure 15) yielding DISCERN and scVI to be the top-performing methods (Table 4). Overall, DISCERN achieved the best performance. The imputation tools are not able to correct batch effects (batch mixing) since they are not developed for this task. Interestingly, no imputation method showed significant improvement in cell type separation.

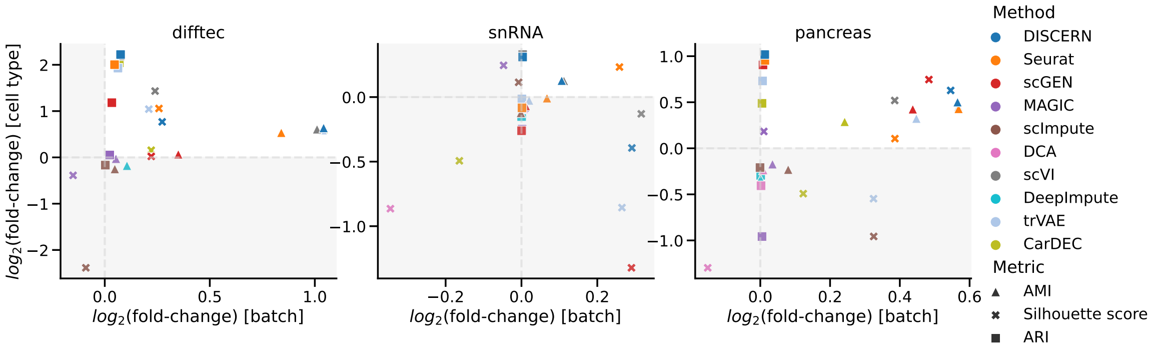


Figure 15: Overview of the batch correction performance results measured by the area under the curve for Silhouette scores (Figure 12), ARI (Figure 13), and AMI (Figure 14) on the pancreas, the difftec and the snRNA-seq data set. Each dot represents a method and evaluation metric. Values were computed as $\log_2$-fold-change improvement compared to the uncorrected data. The grey area indicates worse performance and the grey dotted line indicates equal performance compared to uncorrected data. In the best case, i.e. improved performance compared to uncorrected data, the methods should fall in the quadrant with the white background.

3.1.2 Reference based imputation

Expression reconstruction can partly be seen as expression imputation. Therefore, the ability of these batch correction and imputation models to adjust expression information was evaluated. To test whether these models can reconstruct the cell type-specific gene expression, the correlation of the mean gene expression per cell type was computed on the pancreas, the difftec, and the snRNA-seq data set.

Table 4: Overview of performance in batch correction tasks. The area under the curve for Silhouette scores (Figure 12), ARI (Figure 13), and AMI (Figure 14) are used to rank models for each data set. The sum of ranks across the pancreas, the difftec, and the snRNA-seq are shown. Best values (lowest sum of ranks) are highlighted in bold. If models achieved the same values, the average rank was used, e.g. if two methods perform best, both get a rank of 1.5 assigned. Models are ordered by their overall rank.

	ARI		AMI		Silhouette score		Overall
	batch	cell type	batch	cell type	batch	cell type	
DISCERN	**5.0**	**4.0**	**7.0**	**3.0**	**4.0**	11.0	**34.0**
scVI	6.0	5.0	**7.0**	7.0	8.0	**8.0**	41.0
Seurat	10.0	12.0	9.0	11.0	10.0	9.0	61.0
trVAE	14.0	13.0	11.0	13.0	16.0	17.0	84.0
scGEN	19.0	20.0	16.0	15.0	9.0	16.0	95.0
CarDEC	17.0	15.0	15.0	16.0	21.0	17.0	101.0
MAGIC	16.5	25.5	22.5	20.5	25.5	12.5	123.0
DCA	21.5	20.5	24.5	23.5	29.5	24.5	144.0
scImpute	30.0	25.0	27.0	28.0	20.0	20.0	150.0
DeepImpute	26.0	25.0	26.0	28.0	22.0	30.0	157.0

All methods, except DeepImpute, show a high Pearson correlation (> 0.90) on the difftec data set, whereas the performance on the pancreas and the snRNA-seq data set varies more (Figure 16). On the pancreas and the snRNA-seq data set the deep learning methods, which allow for projection, i.e. DISCERN, scGEN, scVI, and trVAE, achieve the highest Pearson correlation (Figure 16).

The standard deviation across the three data sets after reconstruction was tested, to infer if the models were able to capture the variance present in the data set or predict the mean expression per cell type for all cell types. The latter would be undesirable but still result in a good performance in Figure 16. While DISCERN, scVI, Seurat, scGEN, and scImpute can achieve good correlation values on the pancreas data set (> 0.80), the performance of most methods drops on the more complex difftec and snRNA-seq data set (Figure 17). Especially on the difftec data set, where uncorrected expression data shows good correlation (≈ 0.90), most methods show a decrease in correlation (Figure 17).

Despite the standard deviation, the distribution of (biological and technical) dropout events is important for downstream applications. Therefore, also the correlation of the dropout fraction per gene and cell type was compared after reconstruction. On

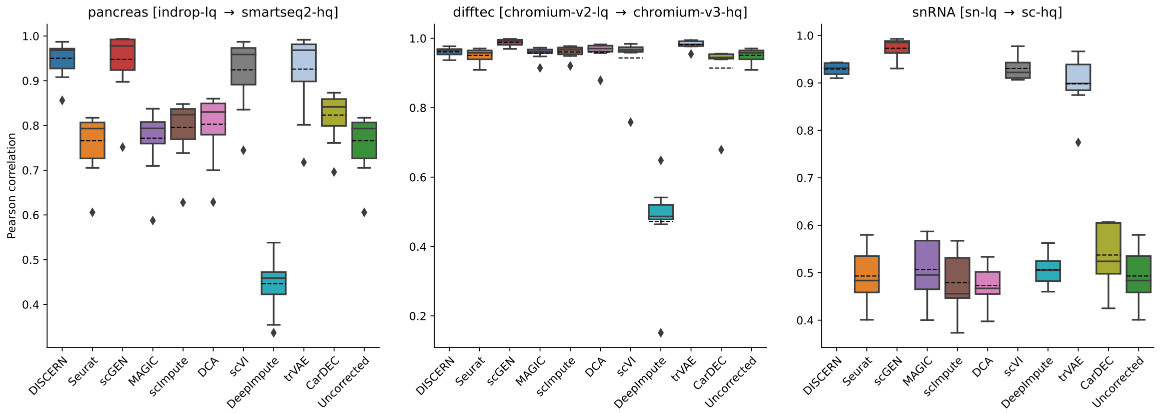


Figure 16: Correlation of the average gene expression in the pancreas, the difftec, and the snRNA-seq data set. From the pancreas data set the indrop-lq was reconstructed using the smartseq2-hq batch. From the difftec data set the chromium-v2-lq batch was reconstructed using chromium-v3-hq batch as a reference and From the snRNA-seq data set the sn-lq batch was reconstructed using the sc-hq batch. Pearson correlation was computed for each cell type and summarized as box plots. The dotted line shows the arithmetic mean. For the pancreas data set 13**, for the difftec dataset eight and for the snRNA-seq data set seven cell types are shown.**

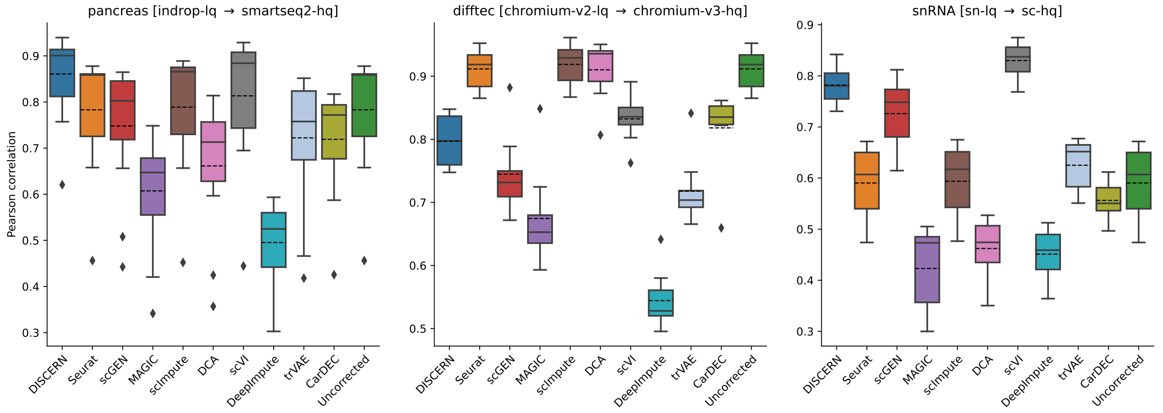


Figure 17: Correlation of the standard deviation of gene expression values in the pancreas, the difftec, and the snRNA-seq data set. From the pancreas data set the indrop-lq was reconstructed using the smartseq2-hq batch. From the difftec data set the chromium-v2-lq batch was reconstructed using chromium-v3-hq batch as a reference and from the snRNA-seq data set the sn-lq batch was reconstructed using the sc-hq batch. Pearson correlation was computed for each cell type and summarized as box plots. The dotted line shows the arithmetic mean. For the pancreas data set 13**, for the difftec dataset eight and for the snRNA-seq data set seven cell types are shown.**

the pancreas data set all methods, except DeepImpute, slightly increase the Pearson correlation compared to uncorrected data. DISCERN shows the strongest median increase of ≈ 0.20 (Figure 18). In contrast, on the difftec data set all methods, except DISCERN, scImpute, and Seurat show a decrease in correlation, whereas DISCERN is the only method showing an increase over uncorrected data. On the snRNA-seq data set, especially the deep learning-based methods are performing better. DISCERN, scGEN, scVI, DeepImpute, and trVAE show improved correlation compared to uncorrected data. DISCERN is the only method achieving a median correlation close to 1.0 for all three data sets (Figure 18).

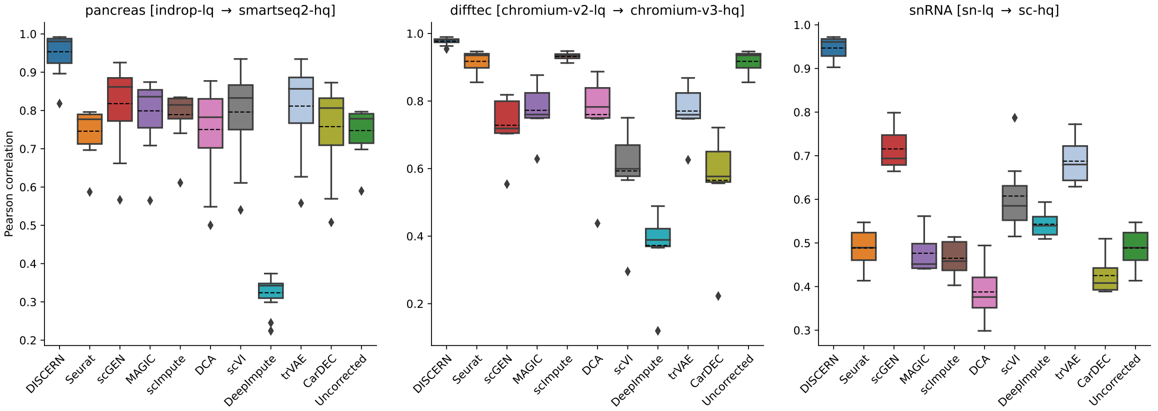


Figure 18: Correlation of dropout values in the pancreas, the difftec, and the snRNA-seq data set. From the pancreas data set the indrop-lq was reconstructed using the smartseq2-hq batch. From the difftec data set the chromium-v2-lq batch was reconstructed using chromium-v3-hq batch as a reference and from the snRNA-seq data set the sn-lq batch was reconstructed using the sc-hq batch. A log-normalized gene expression value below 0.1 was treated as a dropout value. Pearson correlation was computed for each cell type and summarized as box plots. The dotted line shows the arithmetic mean. For the pancreas data set 13, for the difftec data set eight and for the snRNA-seq data set seven cell types are shown.

The evaluation of reference-based reconstruction was performed using a downstream application, i.e. differential gene expression analysis. To detect marker genes for cell types, differential gene expression is typically calculated for one cell type vs all other cell types (one-vs-rest). Top-ranking genes are then considered cell type-specific marker genes. For the low-quality batch, the resulting t-statistics were compared to results obtained using the high-quality batch. This was done using overlapping cell types between high and low-quality data. Since the rank of cell type-specific genes is more interesting than their actual values, the Spearman's rank correlation was used as a metric in this experiment. DISCERN, Seurat, scGEN, MAGIC, scVI, and CarDEC improve the correlation on the pancreas data set compared to uncorrected data (Figure 19). On the difftec data set DISCERN, scGEN, scVI and CarDEC improve the median correlation and on the snRNA-seq data set DISCERN, scGEN, scVI, and trVAE improve the median correlation (Figure 19).

All models and the uncorrected data show a higher variance of correlation values on the pancreas and the snRNA-seq data set (Figure 19), indicating that some cell type marker genes are more similar between batches than others.

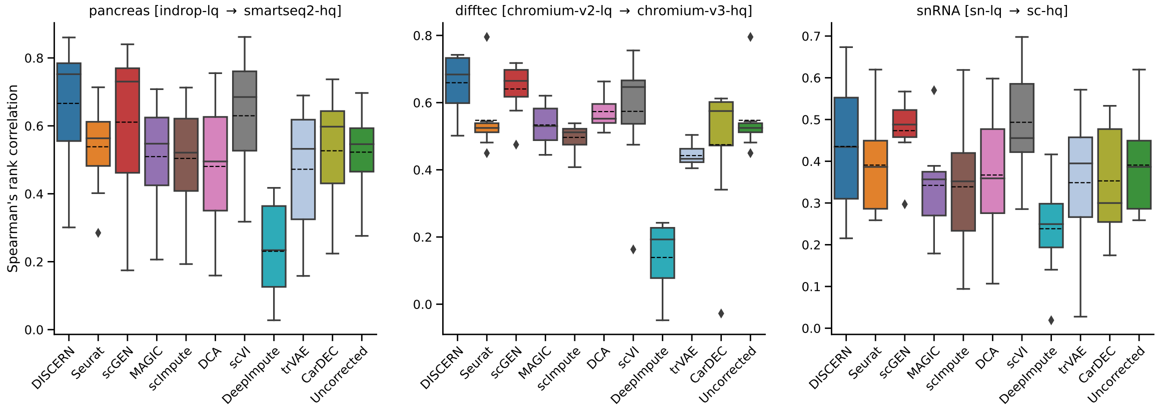


Figure 19: Correlation of t-statistics in one vs rest differential expression analysis of the pancreas, the difftec, and the snRNA-seq data set. From the pancreas data set the indrop-lq was reconstructed using the smartseq2-hq batch. From the difftec data set the chromium-v2-lq batch was reconstructed using chromium-v3-hq batch as a reference and from the snRNA-seq data set the sn-lq batch was reconstructed using the sc-hq batch. Spearman's rank correlation was computed for each cell type and summarized as box plots. The dotted line shows the arithmetic mean. The number of cell types differs between the three methods: 7, 8, 11 cell types are shown for snRNA-seq, difftec and pancreas data sets, respectively.

Thus, these variations are influenced by the number of cells in the pancreas and the snRNA-seq data set, but not in the difftec data set. In the pancreas and the snRNA-seq data set, the achieved correlation values show a linear dependency with the number of cells per cell type independent of the considered quality and without reconstruction (Figure 20). Thus, an increasing number of cells seems to provide a more robust marker gene ranking.

To summarize the performance on the reference-based imputation tasks, mean expression correlation (Figure 16), standard deviation correlation (Figure 17), differential expressed gene (DEG) correlation (Figure 19), and dropout value correlation (Figure 18), and the mean performance across cell types was ranked per data set.

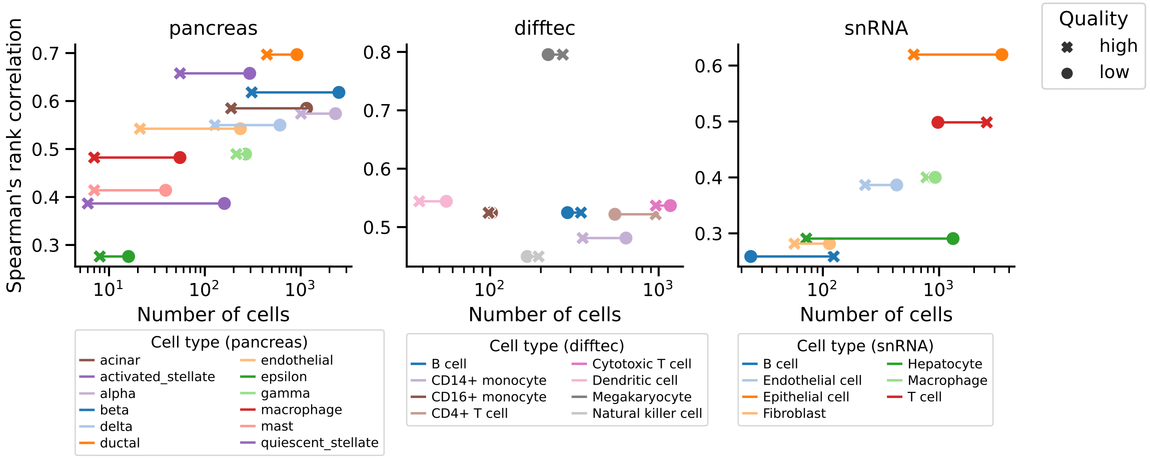


Figure 20: Correlation of t-statistics in one vs rest differential expression analysis without reconstruction (Uncorrected in Figure 19) of the pancreas, the difftec, and the snRNA-seq data set compared to the abundance of cell types. For the pancreas data set indrop-lq batch (low quality) and smartseq2-hq (high quality), for the difftec data set the chromium-v2-lq batch and the chromium-v3-hq and for the snRNA-seq data set the sn-lq batch and sc-hq are shown. The cell type counts for the low-quality (lq) and the high-quality (hq) batches are connected using colored lines.

The rank-sums across data sets are then displayed in Table 5. For the reconstruction of mean expression, scGEN achieves the best results, followed by DISCERN on rank two. For the reconstruction of the standard deviation of gene expression, scVI achieves the best performance, followed by DISCERN and scImpute on rank two. For the reconstruction of dropout, DISCERN achieves rank one, followed by trVAE and scGEN on rank two. On the DEG correlation experiment again DISCERN achieves the best performance, followed by scGEN on rank two. Overall, DISCERN achieves the best performance across all tested evaluation metrics on reference-based expression reconstruction (Table 5).

3.1.3 Simulation based evaluation

Previous experiments utilized a batch available within the data set as ground truth for all comparisons. In this section, experiments using held-out data will be described.

Table 5: Overview of performance for imputation tasks. Mean values for mean expression correlation (Figure 16), standard deviation correlation (Figure 17), DEG correlation (Figure 19), and dropout value correlation (Figure 18) are used to rank models per data set. The sum of ranks across the pancreas, the difftec, and the snRNA-seq are shown. Best values (lowest sum of ranks) are highlighted in bold. If models achieved the same values the average rank was used. Models are ordered by their overall rank.

	Mean expression	Standard deviation	Dropout	DEG	Overall
DISCERN	7.0	9.0	**3.0**	**5.0**	**24.0**
scGEN	**4.0**	15.0	11.0	7.0	37.0
scVI	14.0	**7.0**	17.0	10.0	48.0
Seurat	24.0	12.0	17.0	13.0	66.0
trVAE	9.0	19.0	11.0	27.0	66.0
scImpute	20.0	9.0	16.0	23.0	68.0
CarDEC	19.0	18.0	25.0	14.0	76.0
DCA	21.0	19.0	25.0	17.0	82.0
MAGIC	21.0	28.0	15.0	19.0	83.0
DeepImpute	26.0	29.0	25.0	30.0	110.0

Removal of expression

In the first experiments, the comparison using held-out data was done by creating an artificial data set. The high-quality batch B was split into two parts with a similar number of cells. One part remains unaltered, called B-hq, and from the other part $B \setminus B$-hq the genes determining the cell type were removed. This modified part will be called B-lq. These genes were determined using one-vs-rest differential gene expression analysis and gene set enrichment using the KEGG database [63]. Technically, the expression values of genes involved in the top pathway according to adjusted p-value per cell type were set to zero. For these experiments, only highly abundant cell types were used. This process was applied to the pancreas data set by splitting smartseq2 into smartseq2-lq and smartseq2-hq and to the difftec data set by splitting chromium-v3 into chromium-v3-lq and chromium-v3-hq (Figures 21 and 22).

On the modified pancreas data set, DISCERN, scGEN, scImpute, scVI, DeepImpute, trVAE, and CarDEC achieve the largest mean Pearson correlation ≥ 0.80 while MAGIC, Seurat, and DCA achieve low mean Pearson correlation < 0.80. DISCERN, trVAE and scVI reconstruct the mean expression yielding a correlation of 0.99. DISCERN achieves the lowest standard deviation between cell types, indicating that DISCERNs performance is independent of the cell type (Figure 21).

Similarly, for the difftec data set, MAGIC, scImpute, DeepImpute, and DCA are not able to reconstruct the removed gene expression values. In this experiment, Seurat, DISCERN, scGEN, scVI, trVAE, and CarDEC achieved a correlation > 0.90.

However, for the low abundant cell type Megakaryocytes, Seurat, scGEN, scVI, and CarDEC all overestimate the gene expression and thus achieve a low precision in the reconstruction (Figure 22). For Megakaryocytes, DISCERN and trVAE achieve a mean Pearson correlation of 0.98 and 0.95, respectively, and so it can be concluded that they can reconstruct the gene expression values with high precision (Figure 22).

Furthermore, the correlation of differential gene expression was tested on these modified data sets. The t-statistics were compared using Pearson correlation, since only a subset of the genes, which were removed beforehand, are evaluated. DISCERN and scVI show the best performance for the pancreas data set. In particular, DISCERN achieves the highest mean correlation of 0.88 (Figure 23). MAGIC is not able to reconstruct the mean expression (Figures 21 and 22) and thus no differential gene expression analysis could be performed. On the difftec data set DISCERN outperforms all other tested methods (mean correlation 0.85, Figure 23). In this experiment, Seurat shows the second-best mean correlation of 0.65. Overall DISCERN shows an increase in mean correlation between $5\,\%$ to $30\,\%$.

In addition to the DEA t-statistics, also the correlation of DEA $\log_2$-fold-change was evaluated. While this does not consider the variance across cells, as discussed in Section 1.3, it is commonly used to quantify the degree of change in comparisons across conditions. Pearson correlation was used to compare expected and the calculated $\log_2$-fold-changes. scVI shows the best performance on the pancreas and the difftec data with a mean correlation of 0.50 and 0.54, respectively. Seurat achieves the second-best performance on the pancreas and CarDEC on the difftec data set. DISCERN is on rank four and five, respectively (Figure 24). This can potentially be explained by the underestimation of the expression from low-expressed genes, whereas the top-performing methods slightly overestimate the expression of low-expressed genes (Figures 21 and 22).

The DEA results were furthermore used to perform gene set enrichment based on the KEGG [63] database. This is a commonly used downstream application for data

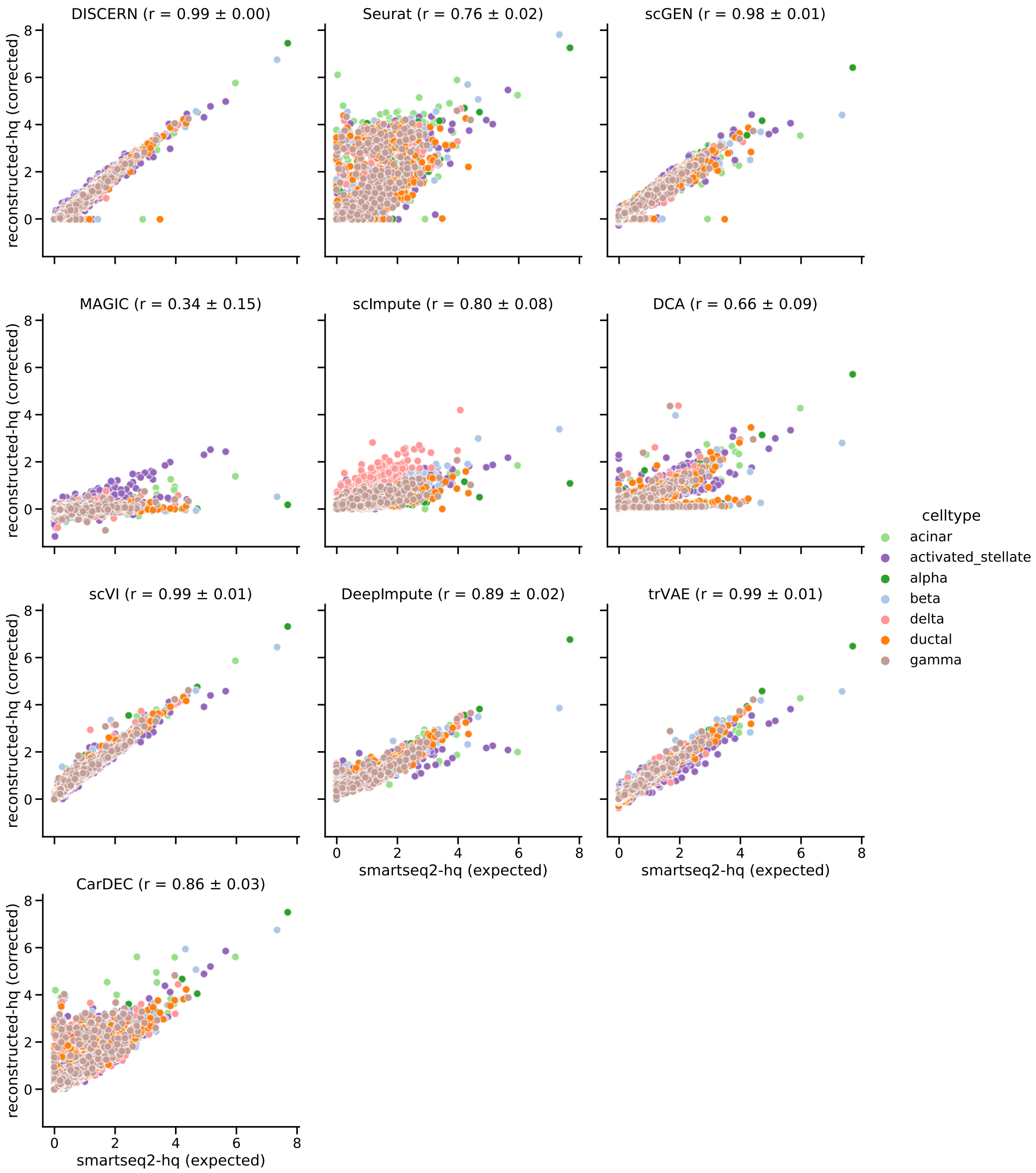


Figure 21: Comparison of the average gene expression reconstruction performance for several methods for the pancreas data set. The data set is based on the pancreas data set where the smartseq2-v3 batch was split into smartseq2-v3-hq and a second part, where selected genes were removed (*in silico* gene dropout). The modified part is called smartseq2-v3-lq. The reconstructed average gene expression (y-axis) is based on the reconstructed or imputed smartseq2-lq data. For DISCERN, scGEN, scVI, and trVAE the projection onto the smartseq2-hq reference is depicted. The expected average gene expression (x-axis) is based on the unmodified smartseq2-lq batch. The mean Pearson correlation with one standard deviation over all cell types is displayed in parentheses of the figure title. Colors indicate the cell type identity.

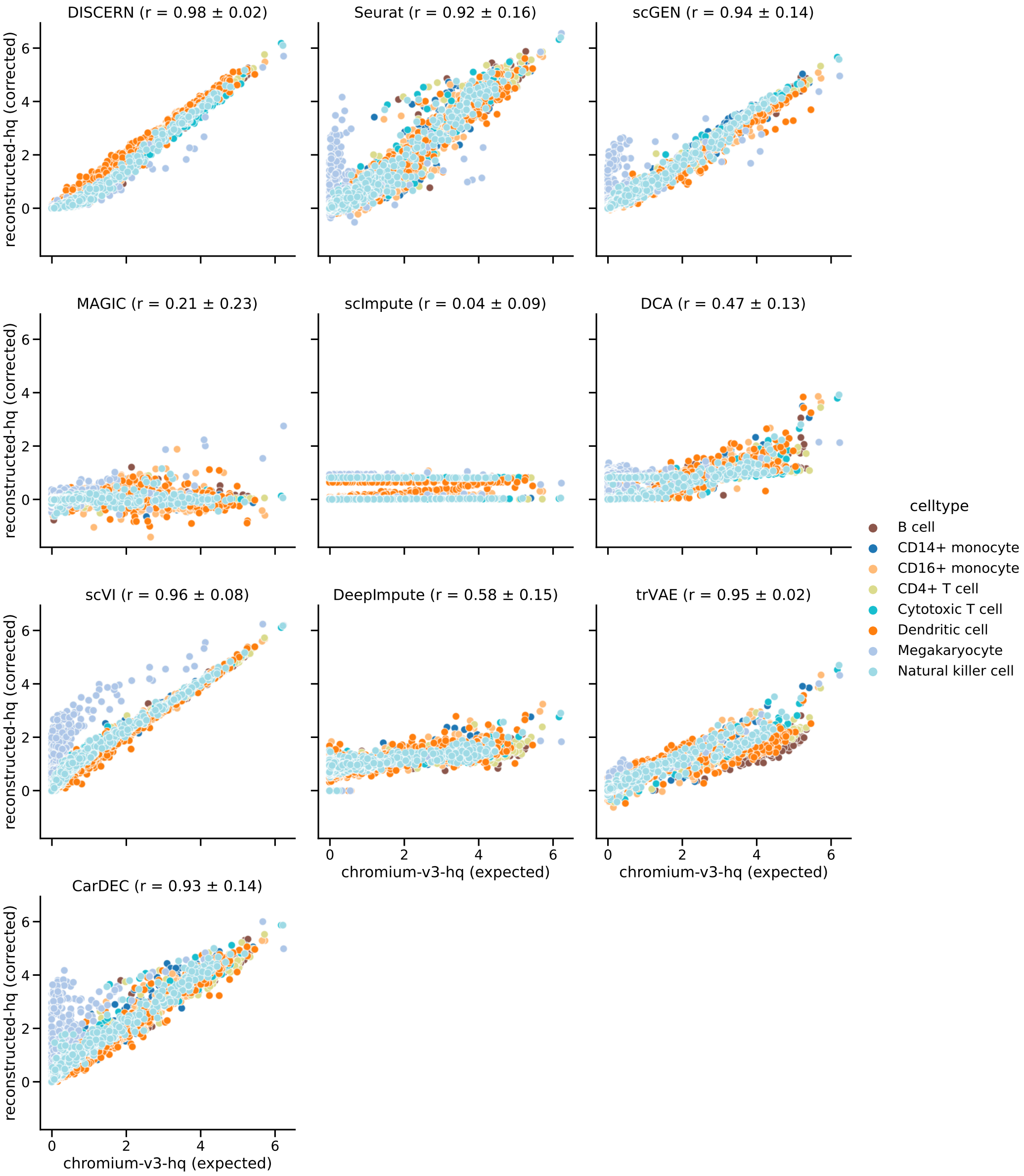


Figure 22: Comparison of the average gene expression reconstruction performance for several methods for the difftec data set. Four imputation (DCA, MAGIC, scImpute, DeepImpute), four batch correction methods (Seurat, scGEN, scVI, trVAE), and DISCERN are compared. The data set is based on the difftec data set where the chromium-v3 batch was split into chromium-v3-hq and a second part, where selected genes were removed (*in silico* gene dropout). The modified part is called chromium-v3-lq. The reconstructed average gene expression (y-axis) is based on the reconstructed or imputed chromium-v3-lq data. For DISCERN, scGEN, scVI, and trVAE the projection onto the chromium-v3-hq reference is depicted. The expected average gene expression (x-axis) is based on the unmodified chromium-v3-lq batch. The mean Pearson correlation with one standard deviation over all cell types is displayed in parentheses of the figure title. Colors indicate the cell type identity.

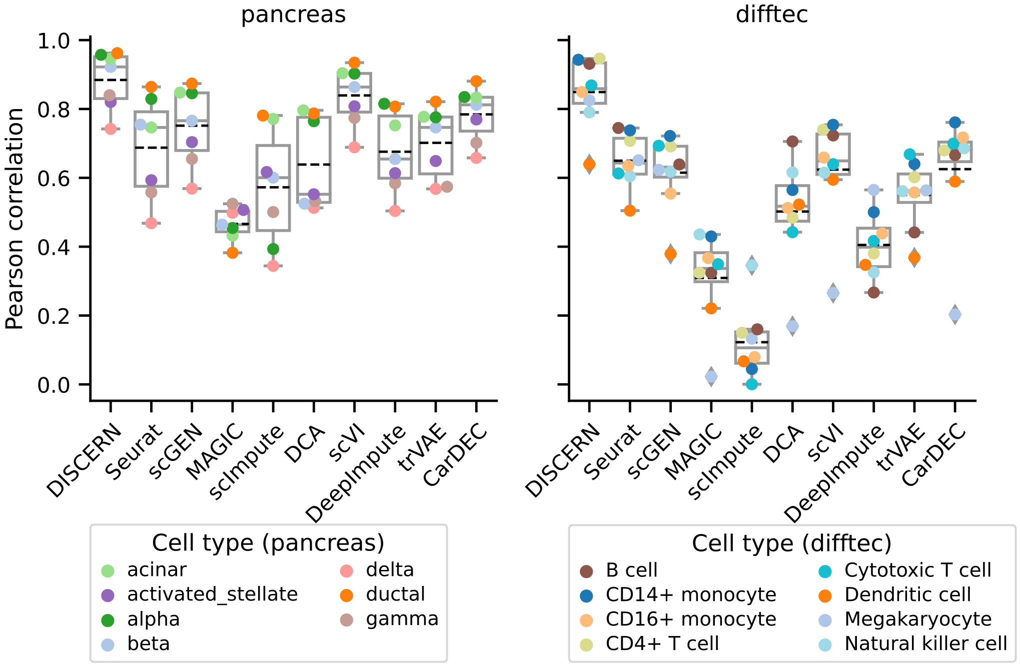


Figure 23: Pearson correlation of DEA t-statistics for a one-vs-rest cell type comparison and in silico gene removal. The data sets are based on the pancreas and the difftec data set where the smartseq2 and the chromium-v3 batch were split into two parts and selected genes were removed from one part, called B-lq. The other part remains unaltered and is called B-hq. The reconstructed DEA t-statistics are based on reconstructed or imputed -lq only, while the expected DEA t-statistics are based on the unmodified B-lq batch. For DISCERN, scGEN, scVI, and trVAE the projection to B-hq is shown. Only genes, which were removed in B-lq were used for calculating the correlation. Colors indicate the cell type identity.

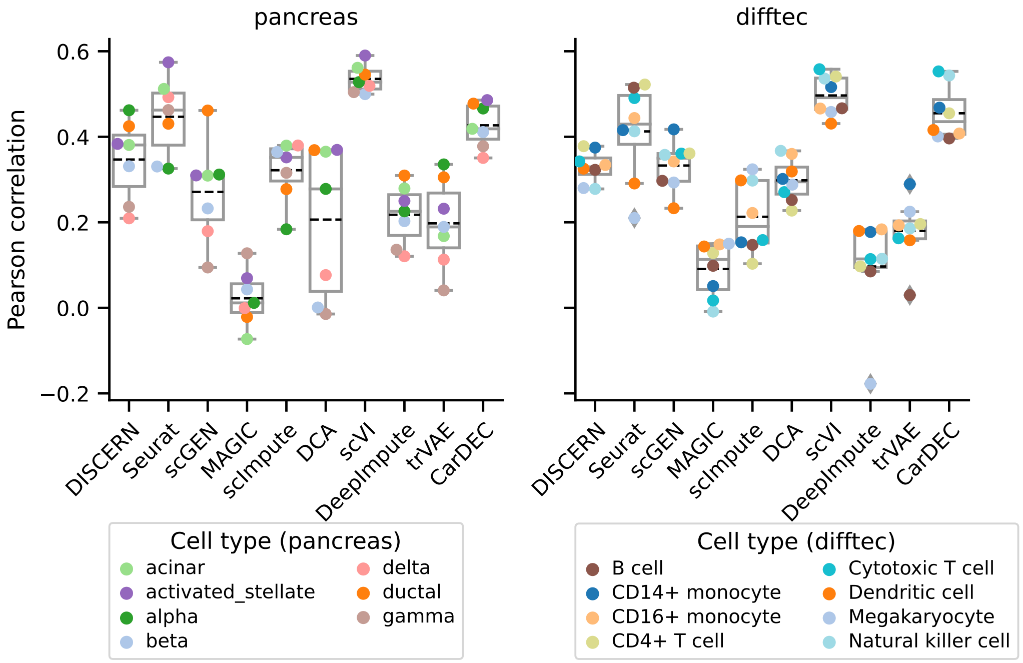


Figure 24: Pearson correlation of DEA $\log_2$-fold-change for a one-vs-rest cell type comparison and in silico gene removal. The data sets are based on the pancreas and the difftec data set where the smartseq2 and the chromium-v3 batch were split into two parts and selected genes were removed from one part, called B-lq. The other part remains unaltered and is called B-hq. The reconstructed $\log_2$-fold-change is based on reconstructed or imputed -lq only, while the expected $\log_2$-fold-change is based on the unmodified B-lq batch. For DISCERN, scGEN, scVI, and trVAE the projection to B-hq is shown. Only genes, which were removed in B-lq were used for calculating the correlation. Colors indicate the cell type identity.

sets of this kind considered here [66, 140].

DISCERN achieved the best performance on the pancreas and the difftec data set with a mean Pearson correlation of 0.86 and 0.75, respectively (Figure 25). On both data sets, scVI achieved the second rank with a mean correlation of 0.80 and 0.65. This partially relates to the correlation ranks of the DEA t-statistics results in which DISCERN and scVI achieve top-ranks. However, Seurat, the second best performing method on the difftec data set for t-statistics reconstruction, does not perform well on the correlation of gene set enrichment scores. This indicates, that Seurat can roughly keep the t-statistics scores, but disturbs the ranking used for gene set enrichment analysis. This does not seem to be the case for scVI and DISCERN.

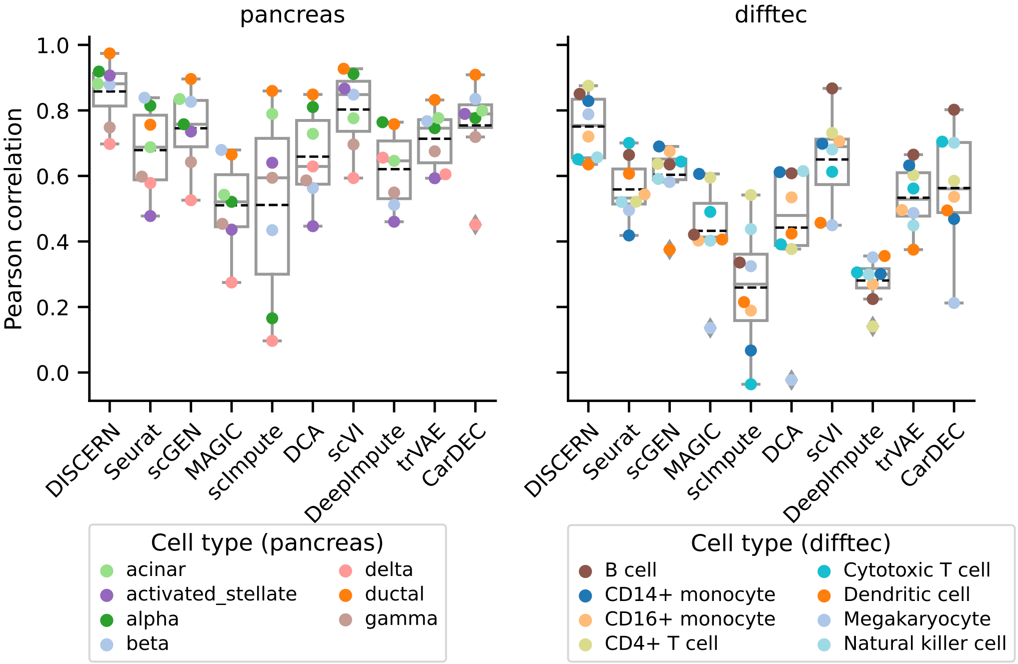


Figure 25: Pearson correlation of KEGG [63] gene set enrichment scores for a one-vs-rest cell type comparison and in silico gene removal. The data sets are based on the pancreas and the difftec data set where the smartseq2 and the chromium-v3 batch were split into two parts and selected genes were removed from one part, called B-lq. The other part remains unaltered and is called B-hq. Instead of directly measuring DEA correlation as in Figures 23 and 24, a gene set enrichment analysis was performed for DEGs. The results were correlated with the ground-truth "expected" information. The reconstructed gene set enrichment scores are based on reconstructed and imputed -lq only, while the expected gene set enrichment scores are based on the unmodified B-lq batch. For DISCERN and scGEN the projection to B-hq is shown. Colors indicate the cell type identity.

In summary, DISCERN achieved the best performance when considering mean expression (Figures 21 and 22), the DEA t-statistics (Figure 23) and the pathway gene set enrichment experiment (Figure 25). scVI ranked best in the DEA $\log_2$-fold-change experiment (Figure 24). Overall DISCERN is the top-ranking method across all conducted experiments involving *in silico* gene removal (Table 6).

Table 6: Overview of performance in the *in silico* gene removal experiments. The mean correlation of the mean expression (Figures 21 and 22), the DEA (Figures 23 and 24) and the pathway (Figure 25) based experiments are used to rank models for each data set. The sum of ranks across the pancreas and the difftec data sets are shown. Best values (lowest sum of ranks) are highlighted in bold. If models achieved the same values, the average rank was used. Models are ordered by their overall rank. $\log_2$-fold-change is abbreviated as $\log_2$-FC.

	Mean expression	DEA (t-statistics)	DEA ($\log_2$-FC)	Pathway	Overall
DISCERN	**2.0**	**2.0**	9.0	**2.0**	**15.0**
scVI	4.0	6.0	**2.0**	4.0	16.0
CarDEC	11.0	6.0	5.0	7.0	29.0
scGEN	8.0	9.0	10.0	7.0	34.0
Seurat	14.0	8.0	5.0	11.0	38.0
trVAE	6.0	11.0	17.0	11.0	45.0
DCA	17.0	15.0	14.0	15.0	61.0
DeepImpute	12.0	15.0	16.0	18.0	61.0
scImpute	17.0	19.0	12.0	21.0	69.0
MAGIC	19.0	19.0	20.0	20.0	78.0

Batch ratio and cell type overlap

In previous experiments, the low-quality and the high-quality batches were modified yielding a similar size, as in the *in silico* gene removal experiments, or kept at their original size, as in the batch correction and imputation experiments (Sections 3.1.1 and 3.1.2). However, the size ratio and cell type overlap between high-quality and low-quality batches is potentially influencing the results. Figure 20 already showed that at least for the pancreas and for the snRNA-seq data sets there is a dependency between the number of cells and the performance of the different tools.

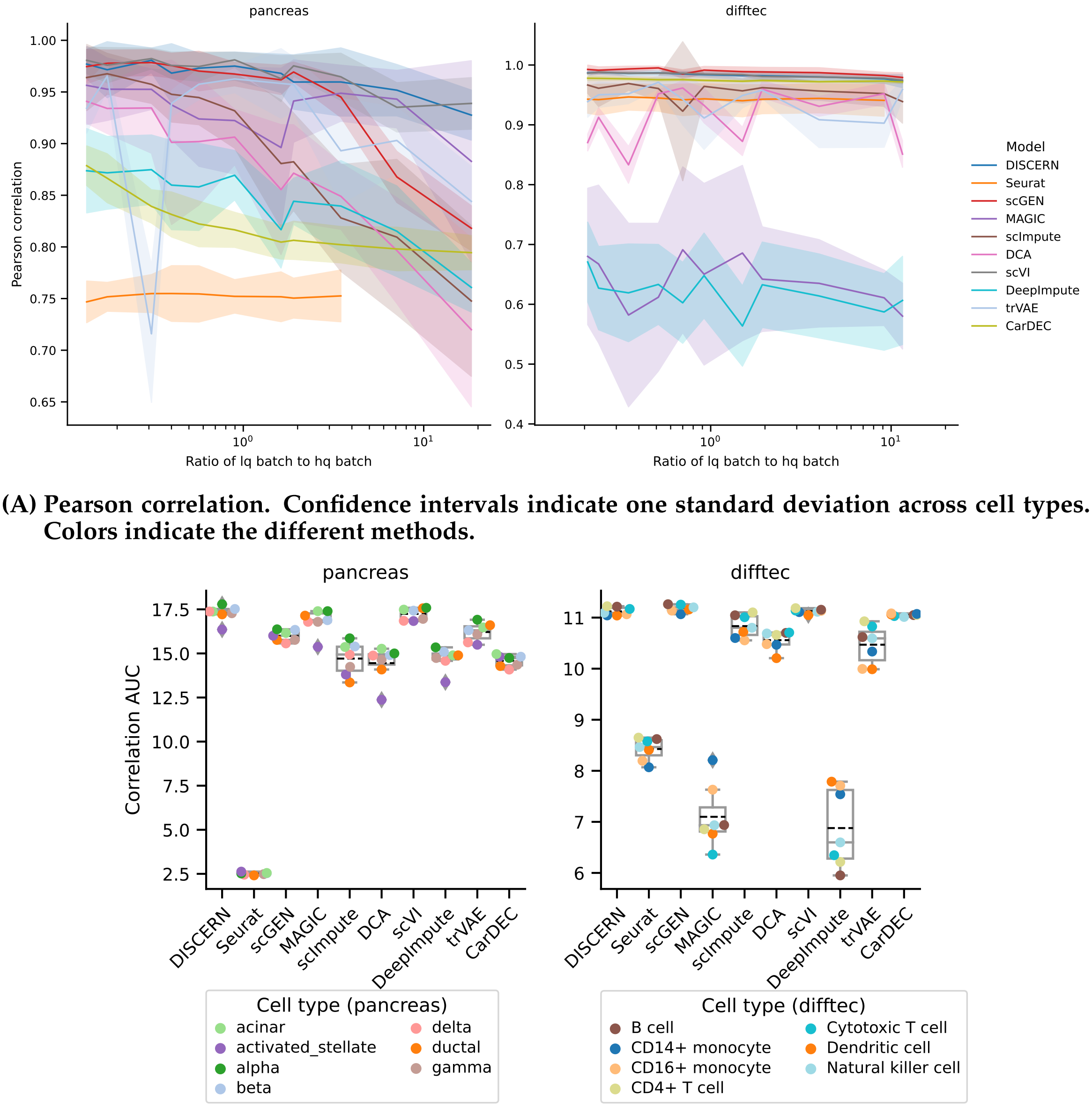
To assess the influence of the batch ratios, the modified data sets from the previous section were used, but the number of cells of the high-quality and low-quality batches was varied. The reconstruction performance was measured using Pearson correlation of the reconstructed mean expression and the expected mean expression from the held-out data. On the pancreas data set scGEN, DISCERN, and scVI achieve the best performance when the high-quality (hq) batch is large compared to the low-quality (lq) batch. All methods, except Seurat, which does not perform well at all, show a drop in performance when the size of the B-hq batch is small compared to the -lq batch. DISCERN and scVI show the least dependency on the batch ratios (Figure 26A). On the difftec data set no such dependency could be observed (Figure 26A). To summarize the graphical representation (Figure 26A), the area under the curves (AUC) was computed for each cell type (Figure 26B). A high AUC corresponds to a high correlation across all investigated ratios of the size of the B-lq batch and the size of the B-hq batch. On the pancreas data set, DISCERN achieves the best mean AUC of 17.3 and scVI achieves the second best with 17.2. On the difftec data set, scGEN, scVI, and DISCERN achieve very similar performance (11.191, 11.126, 11.121). Here scGEN is the best performing method (Figure 26B).

To evaluate the effect of different ratios of B-lq and B-hq sizes on downstream applications, the AUC of Pearson correlation from DEA t-statistics was calculated. On the pancreas data set CarDEC (13.4) performed best, followed by scVI (12.5) and DISCERN (12.1). On the difftec data set DISCERN (8.4) was performing best. CarDEC (7.7) and scVI (7.4) achieved the second and third best performance (Figure 27). Interestingly, with respect to the pancreas data set, all methods achieve the worst performance for activated stellate cells, gamma cells and delta cells. With respect to the difftec data set, all methods achieve the worst performance for dendritic cells.

This seems to be related to the performance of the mean expression reconstruction (Figure 26), where the tested methods obtain similar performance for these cell types.

This is likely not related to the number of cells. Comparing the cell type ratios in the pancreas dataset between the smartseq-lq and the smartseq-hq batch, activated stellate cells (0.86 ± 0.23), gamma cells (1.23 ± 0.73) and delta cells (1.31 ± 0.43) have a similar ratio as cell types for which good performance is achieved, such as ductal cells (1.31 ± 0.43) and acinar cells (0.72 ± 0.28).

To investigate the reconstruction of $\log_2$-fold-changes, the same setting as in Figure 27 was exploited. The resulting $\log_2$-fold-changes were correlated using Pearson correlation and the AUC was calculated. Here scVI and CarDEC were the



(A) Pearson correlation. Confidence intervals indicate one standard deviation across cell types. Colors indicate the different methods.

(B) Area under the curve values for Pearson correlation of mean expression (Figure 26A). Colors indicate the different cell types.

Figure 26: Evaluation of mean gene expression for the pancreas and the difftec data set for different ratios of the size of low-quality (-lq) and the size to high-quality (-hq) training data. The mean gene expression was compared after reconstruction to the B-hq batch. The results were correlated with held-out ground truth. The pancreas and the difftec data set were used as in Figure 23, where the smartseq and the chromium-v3 batch are split into smartseq2-hq, smartseq2-lq, chromium-v3-hq, and chromium-v3-lq.

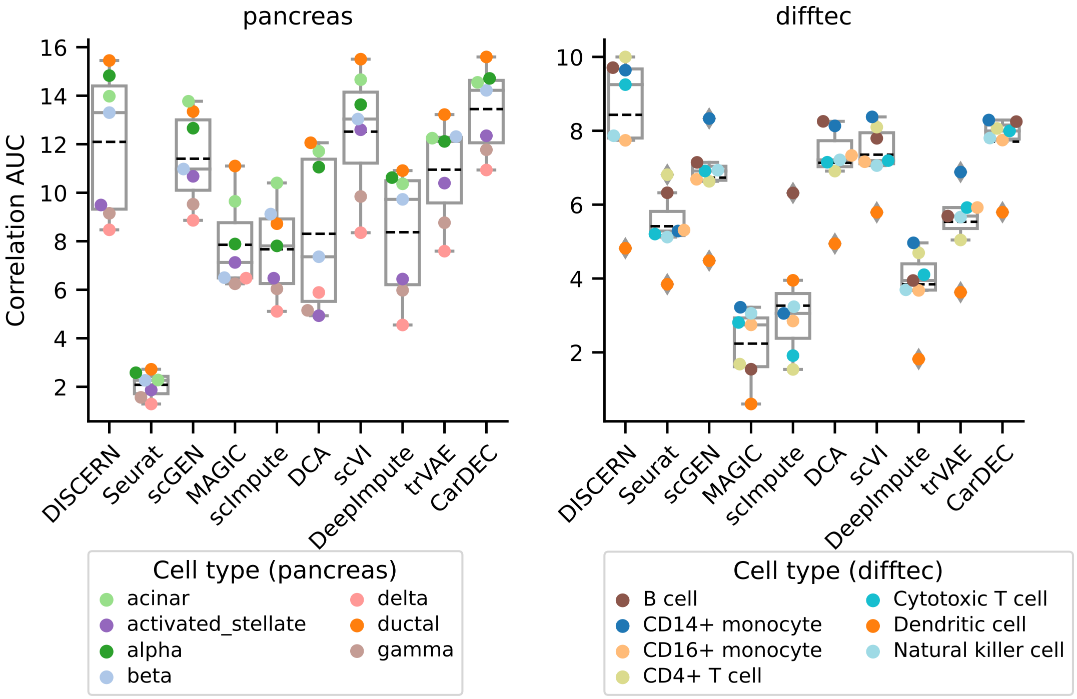


Figure 27: Evaluation of DEA t-statistics for the pancreas and the difftec data set for different ratios of the size of low-quality (-lq) and the size of high-quality (-hq) training data. The t-statistics were compared after reconstruction to the hq batch. The results were correlated with held-out ground truth using Pearson correlation. The area under the curve per cell type is displayed. The pancreas and the difftec data set were used as in Figure 23, where the smartseq and the chromium-v3 batch are split into smartseq2-hq, smartseq2-lq, chromium-v3-hq, and chromium-v3-lq.

best-performing methods. Again, considering the pancreas data set, most methods achieved particularly bad performance for delta and gamma cells.

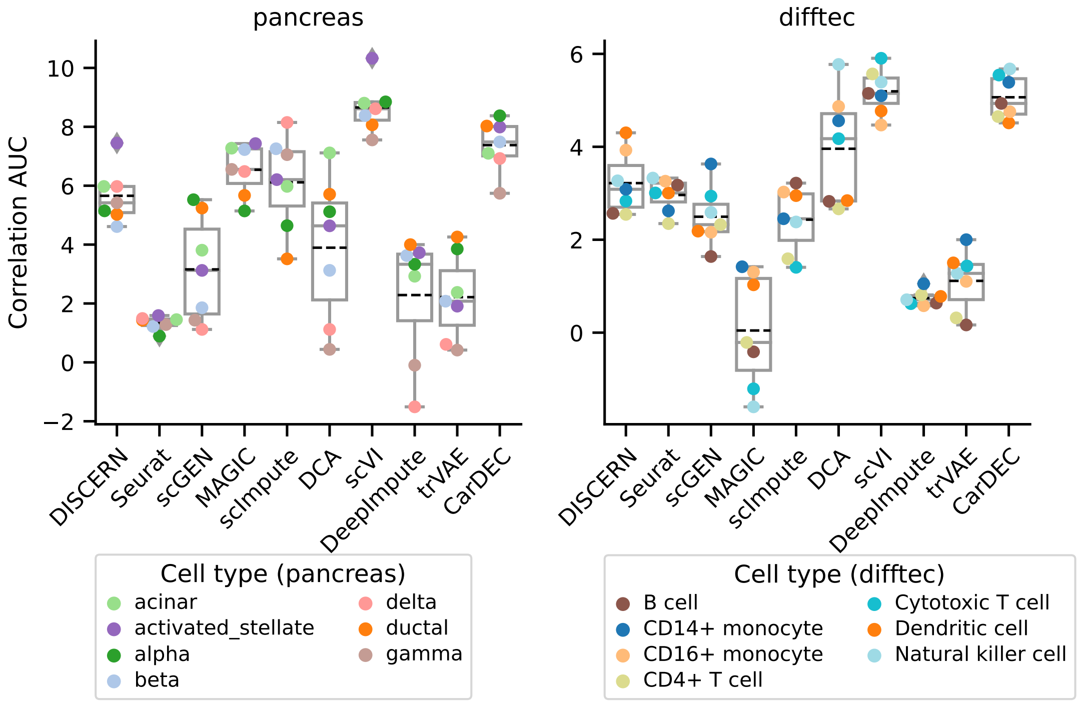


Figure 28: Evaluation of mean gene expression for the pancreas and the difftec data set for different ratios of of the size of low-quality (-lq) and the size of high-quality (hq) training data. The mean gene expression was compared after reconstruction to the hold-out ground truth. The pancreas and the difftec data set were used as in Figure 23, where the smartseq and the chromium-v3 batch are split into smartseq2-hq, smartseq2-lq, chromium-v3-hq, and chromium-v3-lq.

In the previous experiment, the influence of different ratios of the sizes of high and low-quality batches was tested and the cell-type ratios were kept approximately constant. However, depending on the experimental setup, the cell type ratios can be different between sequencing experiments. To mimic this setup, with respect to the pancreas data set, alpha cells were removed from the smartseq2-hq batch. The alpha cells were kept as held-out cells for tests to be used for further evaluation. The reconstruction performance of B-lq alpha cells compared to the held-out -hq alpha cells was assessed using one-vs-rest cell type DEA. This was done using varying numbers of overlapping cell types. Except for the removal of alpha cells, the smartseq-hq was not modified. A varying number of overlapping cell types was achieved by removing cell types from the B-lq batches.

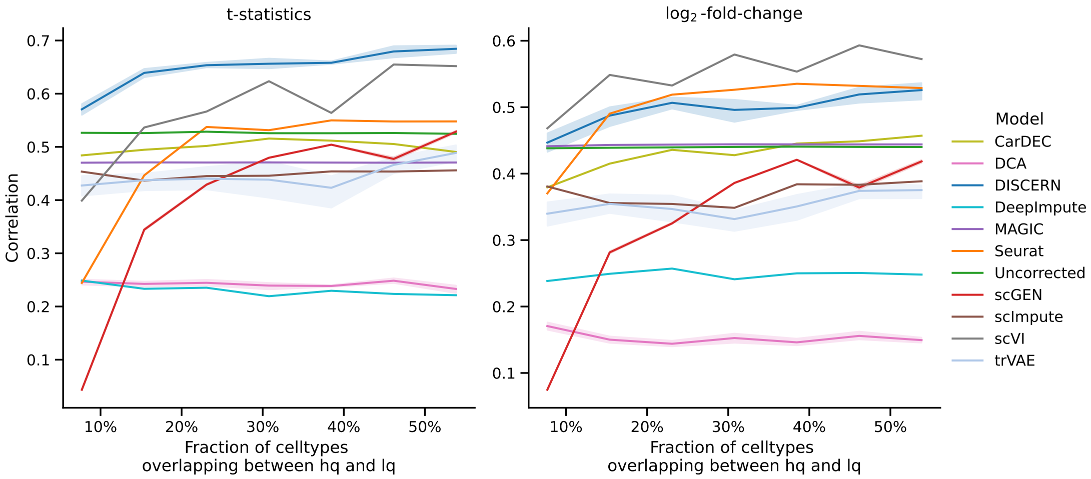


Figure 29: Spearman's rank correlation of the DEA results of alpha cells that were reconstructed and ground-truth alpha cells that were excluded from training, based on the pancreas data set. On the left, the correlation considering the t-statistics is shown. On the right, the $\log_2$-fold-change from the DEA is considered. Different fractions of cell type overlap in the indrop-lq and smartseq2-hq training data were used. Alpha cells were only present in the indrop-lq data and smartseq2-hq alpha cells were extracted as ground truth information. The fraction of cell types, which are non-alpha cells and overlap between low-quality (-lq) and high-quality (-hq) batches are shown on the x-axis. Confidence intervals indicate the standard deviations from five independently trained models.

Considering the t-statistic and Spearman's rank correlation, only DISCERN, scVI, and Seurat were able to show better performance compared to uncorrected data for larger cell type overlap (Figure 29).

DISCERN is the only method showing improvement over uncorrected data for very low overlap ($0\,\%$ to $15\,\%$). When evaluated using the $\log_2$-fold-change again, scVI, Seurat, and DISCERN obtained a higher correlation than uncorrected data. While Seurat reached a slightly better correlation for cell type overlap ($> 20\,\%$), it shows a rapid drop in performance for low overlap ($0\,\%$ to $15\,\%$). For low overlap ranges, only DISCERN and scVI show improved performance (Figure 29).

The reconstructed data of the top-performing methods (Figure 29) including uncorrected data are visualized in Figure 30. Interestingly, all models show unwanted integration of alpha cells from the lq batches and gamma cells from the hq batches. These cell types seem to be very similar. If there is no or low overlap, Seurat yields many clusters composed of more than one cell type, e.g. alpha cells from the smartseq2-hq and acinar cells from the lq batches. This is not the case for scVI and DISCERN. With increasing overlap, the integration task becomes simpler and all models keep most cell types separate (Figure 30).

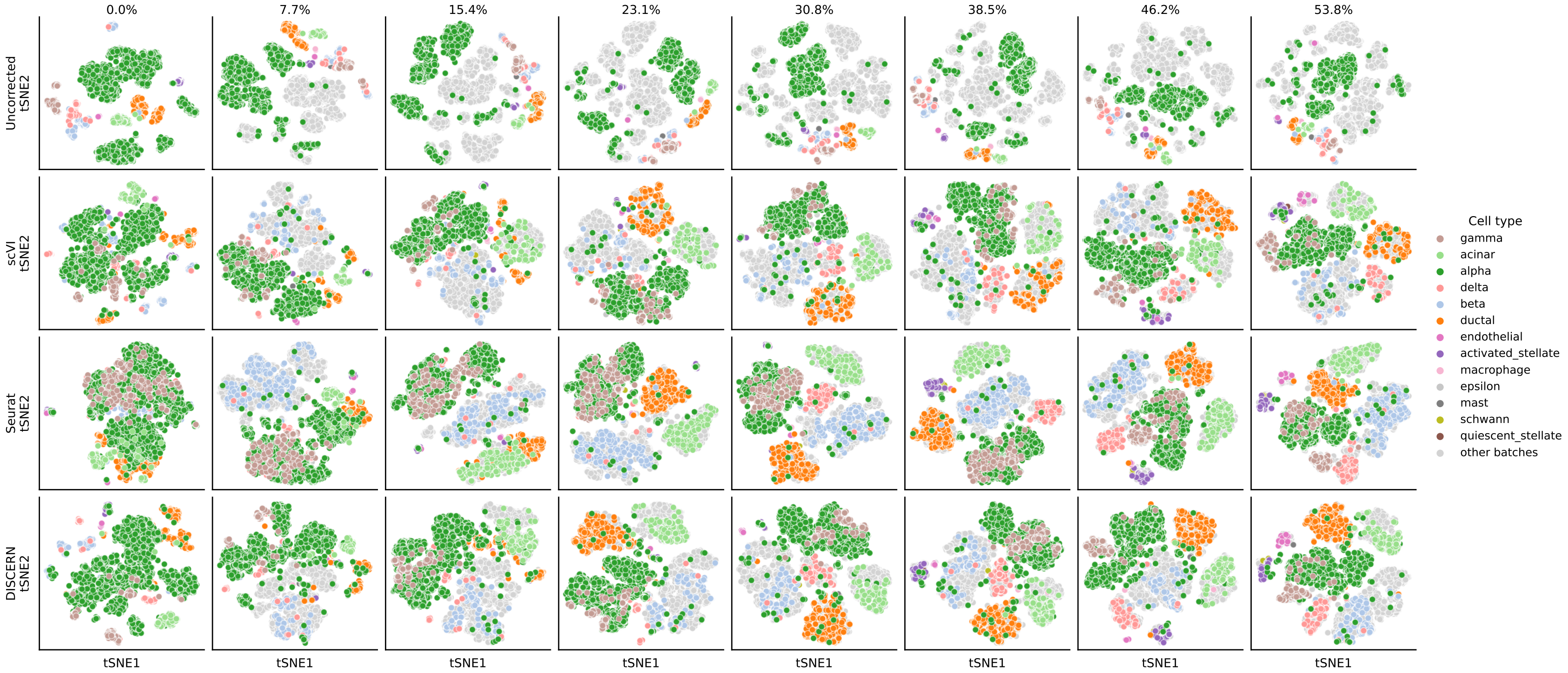


Figure 30: t-SNE of the pancreas data set with removed alpha cells in the smartseq2-hq batch at varying numbers of overlapping cell types. Rows represent the top-performing methods according to Figure 29 and uncorrected data. t-SNE representation was computed on 50 PCA dimensions of scaled data. Cells are colored by cell type. Overlapping cell types from B-lq batches are denoted by light-gray color.

To summarize the results of varying batch sizes and cell type overlap, the ranks of the different tools in the different benchmarks were collected. For the experiments using varying batch sizes, the ranks of the tools for the difftec and the pancreas data set were averaged. scVI obtained the best overall rank and DISCERN achieved the second rank (Table 7).

3.1.4 Summary of Benchmarks

In this section, the evaluation of expression reconstruction using batch correction metrics (Section 3.1.1) and imputation metrics (Section 3.1.2) was described. Furthermore, several *in silico* experiments were done to have available ground truth which is barely available from other sources due to batch effects.

This was done by gene removal and varying sizes of batches and cell type overlap (Section 3.1.3). The sum of ranks, obtained from Tables 4 to 7, were ranked and an overall sum of ranks was calculated to assess overall performance (Table 8). scVI performed best in the batch & cell type ratio experiments, whereas DISCERN performed second best. Especially, the deep learning-based methods, DISCERN, scVI, CarDEC, and the popular Seurat method perform well. The three top-performing methods additionally allow for the selection of a target/reference batch and perform equally well across all tasks. Seurat, however, obtains good performance in the batch correction tasks, but worse performance considering imputed gene expression data and batch/cell type ratios (Table 8). DISCERN achieved the best performance in batch correction, imputation, and gene removal experiments.

3.2 Multi-batch reconstruction

In the previous sections usually, a high-quality batch was selected, considering the number of expressed genes, the mean expression per cell, and the number of cells. However, DISCERN does not require the selection of a reference batch *a priori* and allows the evaluation of multiple references. The tested batch correction tools do not allow for specifying a reference, e.g. Seurat, or incorporate the batch information in the neural network architecture, e.g. scVI and scGEN. Imputation tools do not utilize batch information. Thus, batch correction and imputation tools do not allow for the protection to an average batch or any other modified batch informa-

Table 7: Overview of performance in the batch ratio and cell type overlap experiments. For the batch ratio experiments, the mean correlation of the mean expression (Figure 26) and the DEA (Figures 27 and 28) were considered. For the cell type overlap, the DEA experiment (Figure 29) was considered and the area under the curve was calculated. The sum of ranks across the pancreas and the difftec data sets are shown for the batch ratio experiments for better comparability to the cell type overlap experiments. Best values (lowest sum of ranks) are highlighted in bold. Whenever models achieved the same values, the average rank was used. Models are ordered by their overall rank. $\log_2$-fold-change is abbreviated as $\log_2$-FC and mean expression as mean expr.

	Batch ratio			Cell type overlap		Overall
	Mean expr.	t-statistics	$\log_2$-FC	t-statistics	$\log_2$-FC	
scVI	**2.0**	2.5	**1.0**	2.0	**1.0**	**8.5**
DISCERN	**2.0**	2.0	4.5	**1.0**	3.0	12.5
CarDEC	6.0	**1.5**	2.0	3.0	5.0	17.5
scGEN	3.0	4.5	6.5	8.0	8.0	30.0
MAGIC	6.0	9.0	6.5	5.0	4.0	30.5
Seurat	9.0	8.5	7.5	4.0	2.0	31.0
scImpute	5.5	9.0	5.5	6.0	6.0	32.0
trVAE	5.5	5.5	8.5	7.0	7.0	33.5
DCA	7.5	5.5	4.5	9.0	10.0	36.5
DeepImpute	8.5	7.0	8.5	10.0	9.0	43.0

Table 8: Overall summary of conducted benchmarks. The table shows the obtained rank with the rank sums in brackets. The overall rank is calculated by the sum of ranks in the individual benchmarks. Best values are highlighted in bold. Methods are sorted by their overall rank.

	Batch correction (Table 4)	Imputation (Table 5)	Gene removal (Table 6)	Batch & cell type ratio (Table 7)	Overall
DISCERN	**1.0 (34.0)**	**1.0 (24.0)**	**1.0 (15.0)**	2.0 (12.5)	**5.0**
scVI	2.0 (41.0)	3.0 (48.0)	2.0 (16.0)	**1.0 (8.5)**	8.0
scGEN	5.0 (95.0)	2.0 (37.0)	4.0 (34.0)	4.0 (30.0)	15.0
Seurat	3.0 (61.0)	4.5 (66.0)	5.0 (38.0)	6.0 (31.0)	18.5
CarDEC	6.0 (101.0)	7.0 (76.0)	3.0 (29.0)	3.0 (17.5)	19.0
trVAE	4.0 (84.0)	4.5 (66.0)	6.0 (45.0)	8.0 (33.5)	22.5
MAGIC	7.0 (123.0)	9.0 (83.0)	10.0 (78.0)	5.0 (30.5)	31.0
scImpute	9.0 (150.0)	6.0 (68.0)	9.0 (69.0)	7.0 (32.0)	31.0
DCA	8.0 (144.0)	8.0 (82.0)	7.5 (61.0)	9.0 (36.5)	32.5
DeepImpute	10.0 (157.0)	10.0 (110.0)	7.5 (61.0)	10.0 (43.0)	37.5

tion.

The use of CIN makes it possible to employ the average batch as a reference. This is achieved by multiplying the batch-specific shifting factor by $\frac{1}{N_{batches}}$ and summing over all batch-specific shifting factors, such that Equation (2.1) is modified to:

$$\mathrm{CIN}_{W_\beta}(x) = \gamma\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} + \frac{1}{N_{batches}}\sum_{i=0}^{N_{batches}} W_{\beta_i} \tag{3.1}$$

Reconstruction of all available batches and the average reconstruction can be seen in Figure 31. Qualitatively, all reconstructions yield reasonable cell type clusters (Figure 31, first row) and good batch mixing (Figure 31, second row) regardless of the reference batch. However, considering the fraction of expressed genes, data projected to the high-quality batch, smartseq2 show more expressed genes

for nearly all cells (> 0.2) and reconstructions to low-quality batches, e.g. indrop, show a lower number of expressed genes per cell (< 0.1). Also the fluidigmc1 batch, sequenced using Fluidigm C1, shows a high number of expressed genes, but has a much lower number of cells compared to the high-quality batch and the low-quality batch. Reconstruction to the average batch shows a medium number of expressed genes (0.1 to 0.3) and thus could be used for initial evaluation and is not specific to any reference batch. In this case, $\frac{1}{N_{batches}}$ was used to modify the shifting factor. However, any value could be used, e.g. in specific cases it may make sense to project to the average healthy reference.

3.3 Improving single-cell RNA-seq applications

In the previous sections, a benchmark of DISCERN was performed against several other expression reconstruction methods and the effect of multiple references was evaluated. DISCERN showed improved performance across most evaluated tasks compared to other methods. Therefore, DISCERN will be evaluated for further improvement of downstream applications using a liver metastasis data set, a kidney data set, and a peripheral blood mononuclear cell (PBMC) data set. Furthermore, a case study about COVID-19, investigating the presence of TH17 cell types, was conducted.

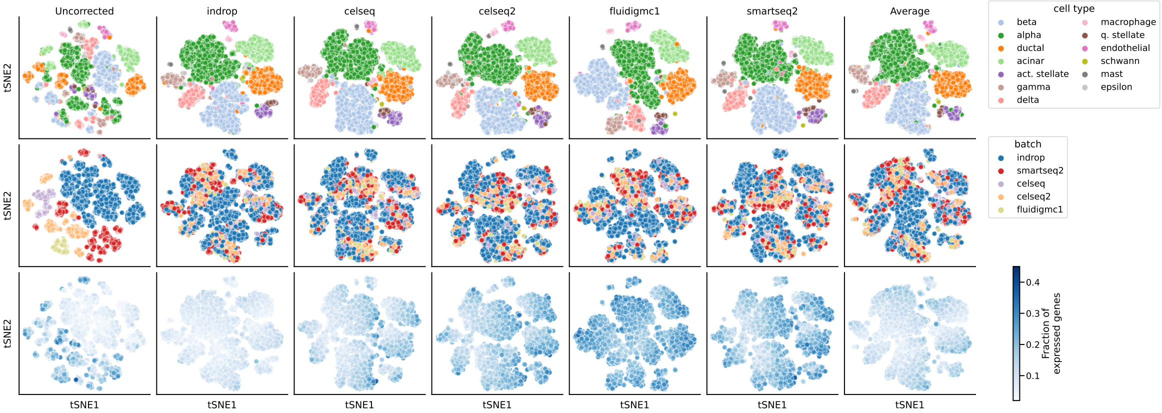


Figure 31: t-SNE representation of all possible reference batch selections after reconstruction using DISCERN, colored by cell type (first row), batch (second row), and the number of expressed genes, i.e. genes with an expression value ≥ 0.1 (last row). Uncorrected data indicate no reconstruction and "Average" uses the average batch information by multiplying the shifting factor by $\frac{1}{N_{batches}}$. Especially the low-quality batches (e.g. indrop) show a low number of expressed genes in uncorrected data (first column). After reconstruction with the indrop batch (second column) a low number of expressed genes can seen for all batches. Reconstruction with a high-quality batch as reference (e.g. smartseq2 in the second last column) a high number of expressed genes can be detected. A medium number of expressed genes can be achieved by using an "Average" batch (last column).

3.3.1 Improving cell type-specific marker gene detection

Pathway expression in snRNA-seq data

The activation of cell type-specific pathways was investigated in the snRNA-seq data set. The snRNA-seq data set was prepared from the same liver metastasis biopsy using scRNA-seq (sc-hq) and snRNA-seq (sn-lq) [45]. The sn-lq batch contains less expression information, measured by the average counts per cell, compared to the sc-hq batch (Table 11). However, snRNA-seq is still often used when intact single cells cannot be recovered from tissue (e.g. after tissue fixation or freezing). It is important to note that nuclear transcripts reflect current transcription activity, which might not correlate well with protein abundance if the transcripts have high lifetimes or transcription rates. On this data set, qualitatively most methods, e.g. Seurat, scGEN, trVAE, and CarDEC, show bad integration performance and are not able to form one cluster for each cell type (Figure 10). Only DISCERN and scVI can create cell type-specific cluster (Figure 10A), but still show a batch effect (Figure 10B). Since they are visually heavily impacted by the batch effect, this data set allows interesting investigation of batch-effect related differences on a gene and pathway level.

The TCR signaling pathway (Figure 32) and the antigen presentation and processing pathway (Figure 33) from KEGG [63] were considered in T cells and macrophages. These cells were annotated in the sn-lq and the sc-hq batch. T cells usually show high expression of the TCR signaling pathway. In contrast, macrophages were used as a negative control, since they do not express the genes from the TCR signaling pathway. However, macrophages show strong expression of the antigen presentation and processing pathway, which is not expressed by T cells used as a negative control. DISCERN, Seurat, DeepImpute, scImpute, CarDEC, scGEN, scVI, DCA, MAGIC, and trVAE were evaluated for their performance in reconstructing the gene expression of these pathways from the sn-lq batch with the sc-hq batch as a reference.

DISCERN shows the most similar expression according to the hierarchical clustering of reconstructed sn-lq, called DISCERN-hq, with the high-quality data (sc-hq) in the TCR signaling pathway in T cells, whereas the imputation methods DeepImpute, CarDEC, DCA, MAGIC, and several batch correction methods, i.e. scGEN and trVAE, are increasing the mean expression values for nearly all genes to higher levels as the reference sc-hq batch and hence introduce more noise (Figure 32A).

Similarly, these methods also increase the expression of several genes involved in the TCR signaling pathway in macrophages and remove cell type-specific expression patterns (Figure 32B). Nearly none of these genes show a high expression in either the sn-lq or the sc-hq data (Figure 32B).

The same uncorrected reconstruction was observed in the antigen presentation and processing pathway in macrophages and T cells (Figure 33). DISCERN reconstructed sn-lq data, called DISCERN-hq, shows a similar expression pattern as the high-quality data (sc-hq).

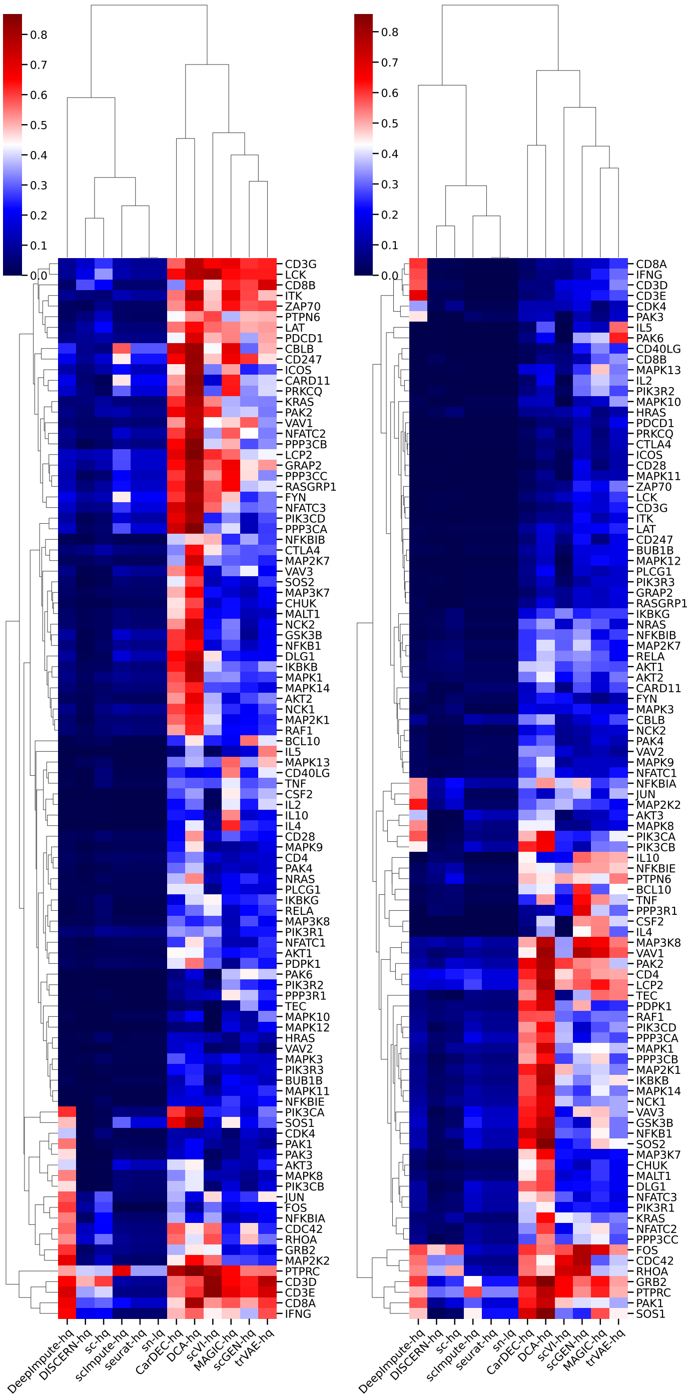
For example, high *CD4* expression is imputed in macrophages, which is not present in either sn-lq or sc-hq, and is not expected in macrophages (see [145]). While for several genes, e.g. *TAP1* and *TABBP*, a high expression in antigen-presenting cells, such as macrophages, is expected [146], this is not found in sn-lq or sc-hq. This discrepancy impedes evaluations based only on prior knowledge.

Cell type marker detection in pancreas

In further experiments, the ability of DISCERN to improve cell type-specific marker gene detection considering multiple sequencing technologies were analyzed. Cell type-specific marker genes are genes that are, possibly together with other genes, exclusively expressed in cells of a single cell type. Therefore, these genes are often used for cell type annotation of scRNA-seq data sets. However, the expression of these marker genes can be low or difficult to detect due to dropout and batch effects. Thus, the pancreas data set was selected to cover a multitude of different sequencing technology-related batch effects, since it contains pancreas cells, which were sequenced using five different sequencing technologies.

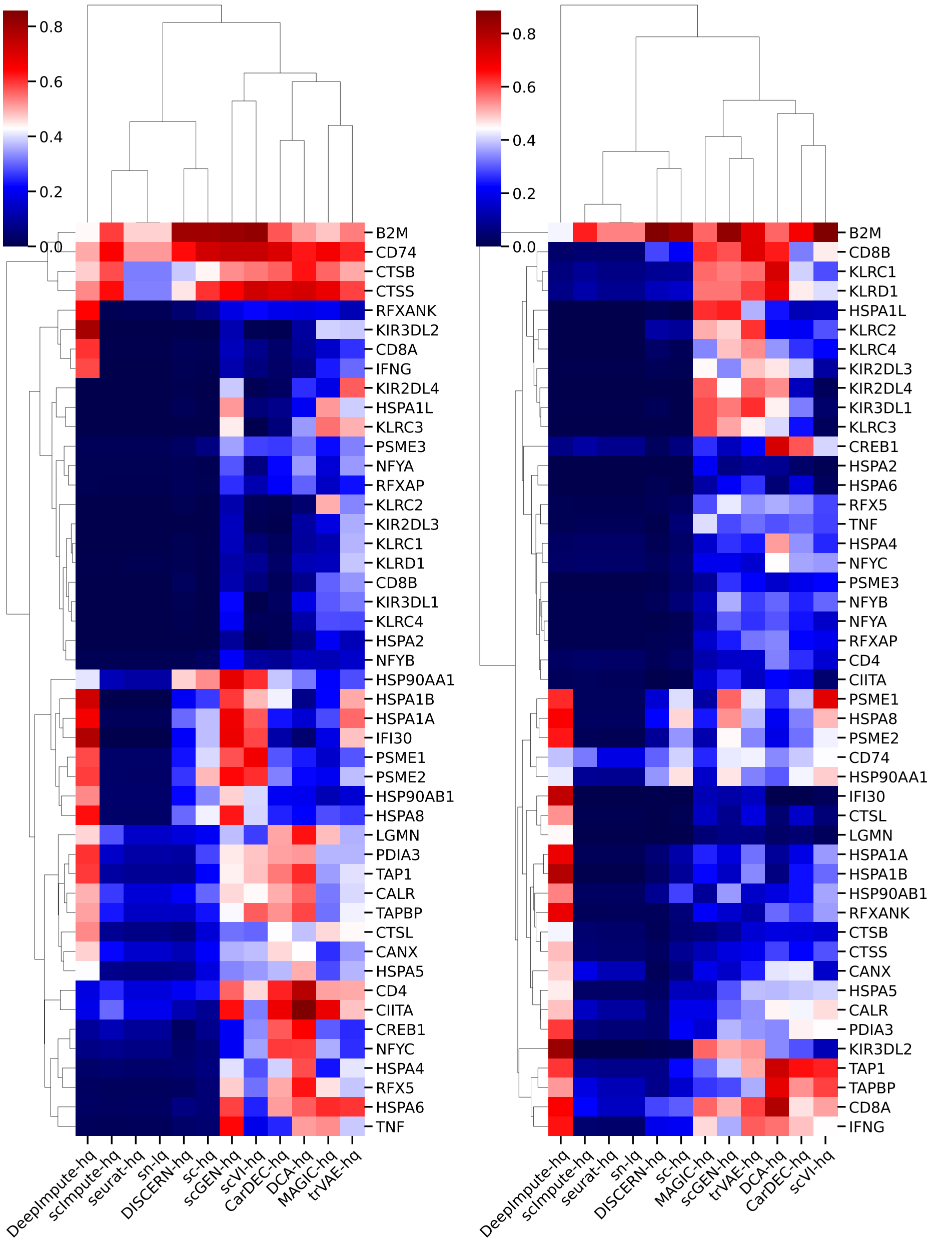
The effect of reconstruction of three cell type-specific genes using DISCERN in the pancreas data set was tested (Figure 34). The expression of insulin (*INS*) in the pancreas is known to be largely restricted to beta cells [147]. This can be observed in the pancreas data set in most batches without reconstruction.

Only the indrop-lq batch shows a diffuse pattern of insulin expression across cell types (Figure 34A, left panel). The diffuse *INS* expression is removed with DISCERN by reconstruction using the smartseq2-hq batch as a reference (Figure 34A, middle panel), yielding a reconstructed expression matrix with *INS* expression spe-



(A) T cells. **(B) Macrophages.**

Figure 32: Average gene expression of TCR signaling genes in T cells (A) and Macrophages (B). The columns show the data in the snRNA-seq (before reconstruction, sn-lq) and snRNA-seq dataset after reconstruction with DISCERN, Seurat, DeepImpute, scImpute, CarDEC, scGEN, scVI, DCA, MAGIC, and trVAE and in the scRNA-seq data (sc-hq). The average expression was min-max scaled after adding a pseudo-count of 1×10^{-3}. The reconstructed-hq shows high similarity with the expression in the sc-hq dataset. Only genes with a maximum expression greater than 0.2 are shown. The gene set was obtained from [144].



(A) Macrophages.

(B) T cells.

Figure 33: Average gene expression of antigen presentation and processing genes in Macrophages (A) and T cells (B). The columns show the data in the snRNA-seq (before reconstruction, sn-lq) and snRNA-seq dataset after reconstruction with DISCERN, Seurat, DeepImpute, scImpute, CarDEC, scGEN, scVI, DCA, MAGIC, and trVAE and in the scRNA-seq data (sc-hq). The average expression was min-max scaled after adding a pseudo-count of 1×10^{-3}. reconstructed-hq shows high similarity with the expression in the sc-hq dataset. Only genes with a maximum expression value greater than 0.2 are shown. The gene set was obtained from [145]. *CD4* and *CD8A* genes are part of the antigen presentation and processing pathway but are not expressed in macrophages. Thus in A, it is not expected, that these genes are expected (see [145]).

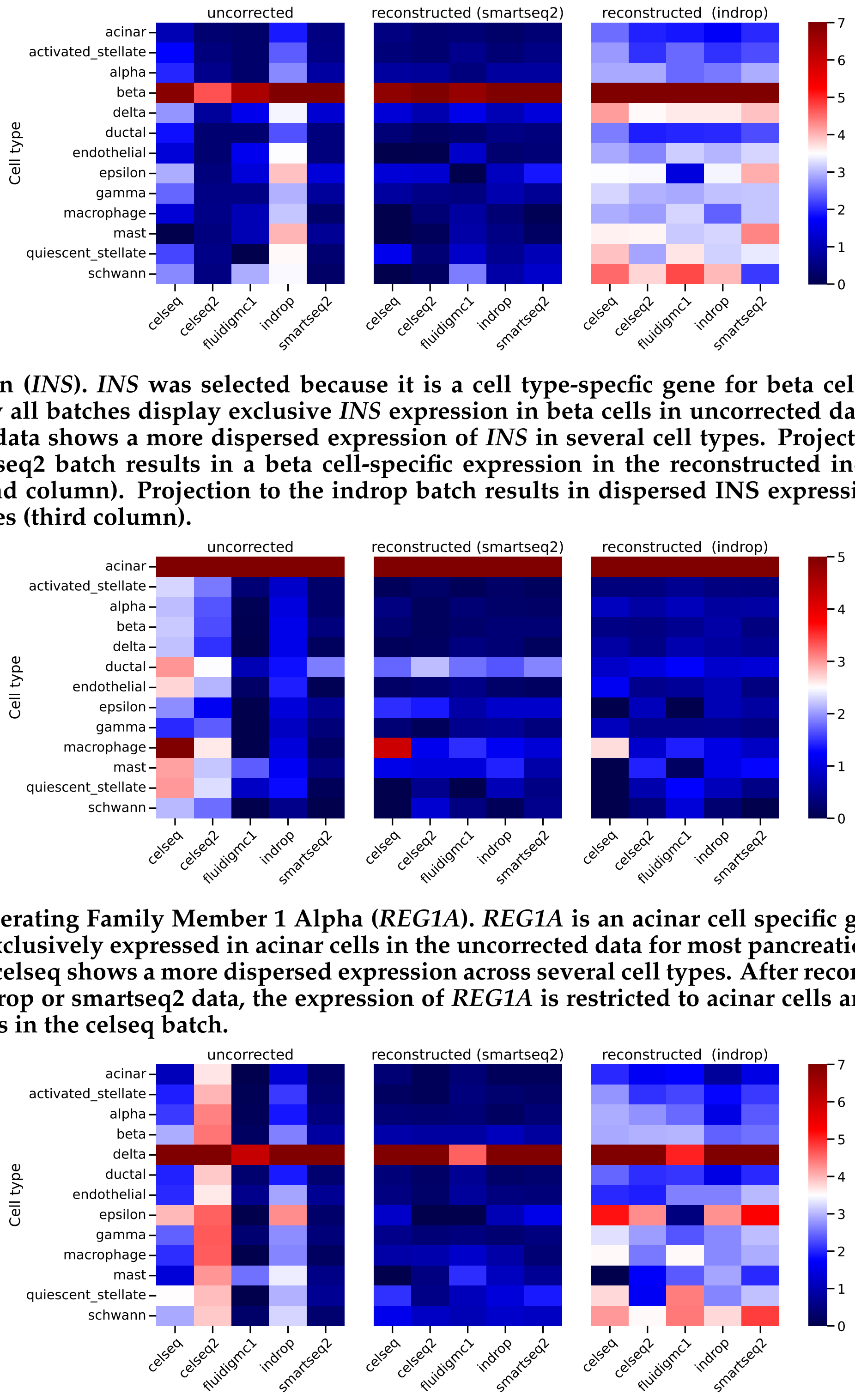
cific to beta cells. However, when selecting the indrop batch as a reference, the diffuse pattern of *INS* expression is transferred to and becomes visible in all batches (Figure 34A, right panel).

A similar result can be observed for the gene Regenerating Family Member 1 Alpha (*REG1A*), which is specifically expressed in acinar cells and was shown to be involved in acinar cell carcinoma [148]. As expected, in the data set *REG1A* is highly expressed in acinar cells. However, the celseq and the celseq2 batch show a diffuse pattern of *REG1A* expression in nearly all cell types (Figure 34B, left panel). A reconstruction using the smartseq2 and the indrop batch as reference does not show such a diffuse pattern of *REG1A* (Figure 34B middle & right panel). The high expression of *REG1A* in macrophages in the celseq batch is likely due to a wrong annotation, because in the celseq batch only one cell was annotated as macrophage and this cell is clustering similarly to the other acinar cells. Since DISCERN has no cell type information available, it correctly retains the high expression of *REG1A* in this cell (Figure 35). For delta cells, Somatostatin (*SST*) was found to be cell type-specific in the pancreas [149]. This can also be seen in the uncorrected data in all batches, especially the smartseq2 and the fluidigmc1 batch (Figure 34C, left panel). Again, the celseq2 batch and the indrop batch show a diffuse expression pattern, which, depending on the reference batch is not present (Figure 34C middle panel) or is increased (Figure 34C right panel).

T cell detection in kidney injury data

To further evaluate the ability of DISCERN to improve cell type detection, a kidney data set of nine patients with acute kidney injury was used. The raw and preprocessed versions of the data set were provided by Rajasree Menon from the group of Matthias Kretzler, University of Michigan [150]. The preprocessing was done using Seurat (version 3) using reciprocal PCA for integration. Unprocessed data is available from the Kidney Precision Medicine Project [151] (see Table 10 for individual patient IDs).

The nine patients were sequenced with scRNA-seq (kidney-hq) and snRNA-seq (kidney-lq), such that 18 samples were obtained in total. After reconstruction of the uncorrected data using DISCERN and the kidney-hq as the reference, one patient (“3210003”) was removed due to very high batch effects and low quality of this patient.



(A) Insulin (*INS*). *INS* was selected because it is a cell type-specfic gene for beta cells. While nearly all batches display exclusive *INS* expression in beta cells in uncorrected data, the indrop data shows a more dispersed expression of *INS* in several cell types. Projection to the smartseq2 batch results in a beta cell-specific expression in the reconstructed indrop data (second column). Projection to the indrop batch results in dispersed INS expression for all batches (third column).

(B) Regenerating Family Member 1 Alpha (*REG1A*). *REG1A* is an acinar cell specific gene [148]. It is exclusively expressed in acinar cells in the uncorrected data for most pancreatic datasets. Only celseq shows a more dispersed expression across several cell types. After reconstruction to indrop or smartseq2 data, the expression of *REG1A* is restricted to acinar cells and macrophages in the celseq batch.

(C) Somatostatin (*SST*). *SST* is known to be produced by delta cells in the pancreas [149], which can be observed for instance in the smartseq2 batch. After reconstruction to the smartseq2 batch, delta cell-specific expression of *SST* is observed for all datasets.

Figure 34: Average gene expression of Insulin (A), Regenerating Family Member 1 Alpha (B), and Somatostatin (C) by cell type (rows) and by batch (columns) in the pancreas dataset. The first column shows the uncorrected datasets, while the second and third column show projections using DISCERN to the smartseq2 and the indrop dataset, respectively.

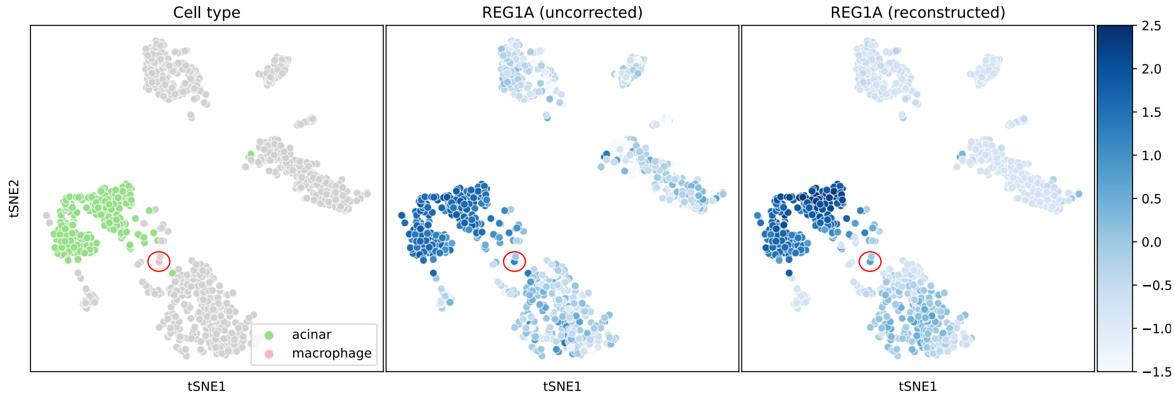


Figure 35: t-SNE representation of the celseq batch from the pancreas data set. The left panel is colored by the cell types acinar and macrophage. The middle and right panels show the scaled expression of *REG1A* in the uncorrected data and after reconstruction with DISCERN using the smartseq2 batch as a reference. The single cell annotated as macrophage is highlighted using a red circle. The t-SNE representation was computed on the top 50 principal components of scaled data.

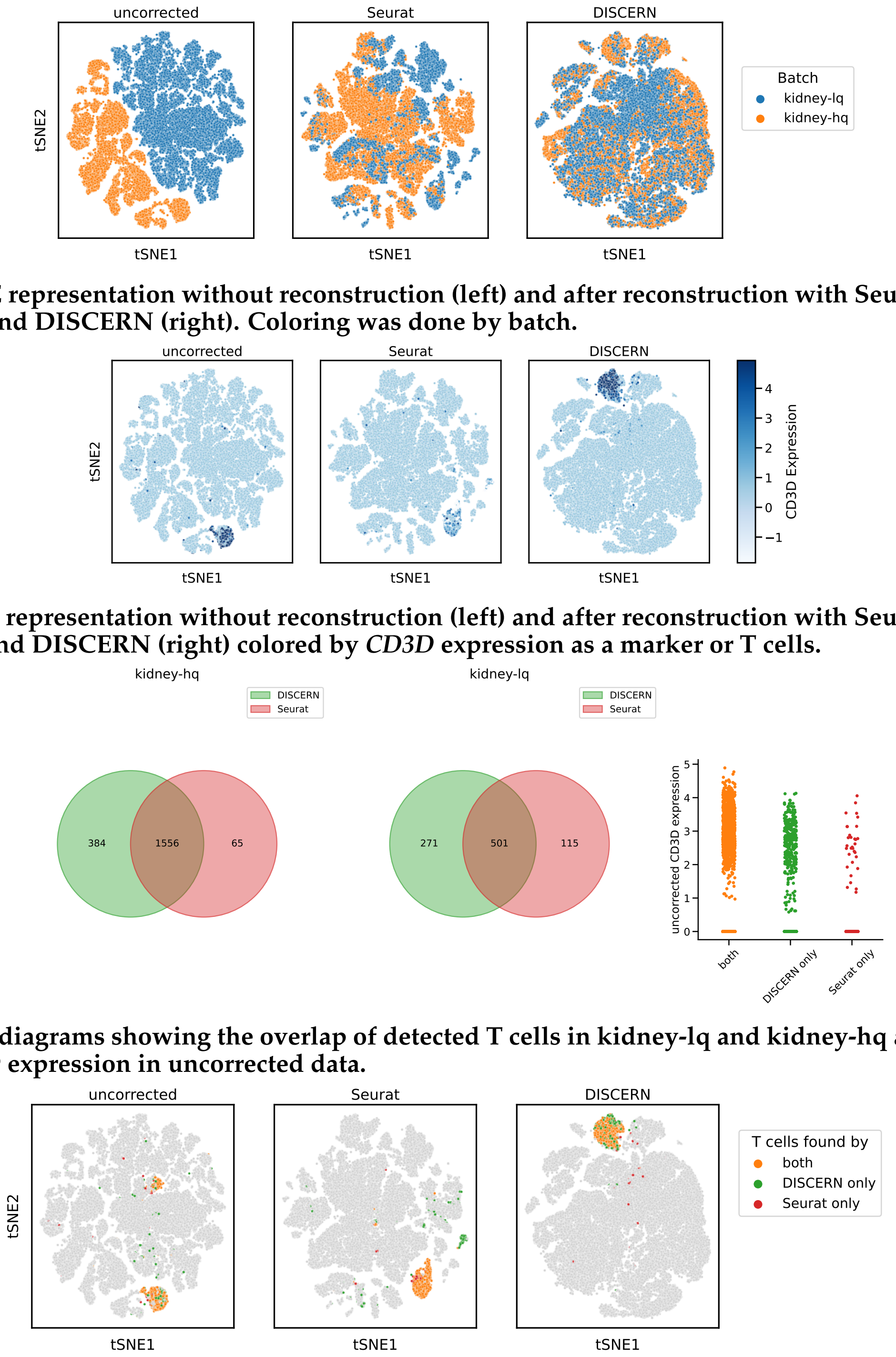
In this data set, the detection of T cells was challenging due to the low number of *CD3D* expressing cells in the kidney-lq batch [150]. *CD3D* is a marker for T cells. However, the total number of cells is larger in the kidney-lq batch (52 934) compared to the kidney-hq batch (29 767). The data set shows a batch effect, which Seurat can only partially remove (Figure 36A). DISCERN shows a stronger integration than Seurat (Figure 36A). Since no ground truth cell type information is available, the cell type annotations could not be used to assess the preservation of biology in data reconstructed by DISCERN and by Seurat. However, the uncorrected data set and the two data sets reconstructed by DISCERN and by Seurat show a cluster of cells, which are highly expressing *CD3D* (Figure 36B). The highest expression is observed in the uncorrected data set and in the data set reconstructed by DISCERN. However, the uncorrected data show only a cluster expressing *CD3D* in the kidney-hq batch, but not in the kidney-lq batch (Figure 36B, right panel). Using Leiden clustering, this cluster of cells expressing *CD3D* in the reconstruction by Seurat and DISCERN was selected and analyzed further. Most of the cells in this cluster are present in the reconstruction by DISCERN and in the reconstruction by Seurat (1 556 in kidney-hq and 501 in kidney-lq). As expected the majority of T cells are detected in the kidney-hq data (Figure 36C, left & middle panel). DISCERN and Seurat can uniquely identify 655 and 220 T cells which were not identified in the reconstruction by the other tool.

To verify that the detected cells in Seurat and DISCERN reconstruction are *CD3D* positive without integration, their expression was displayed (Figure 36C, right panel). T cells in the data reconstructed by DISCERN only show similar *CD3D* expression compared to T cells, which were commonly detected in DISCERN and Seurat reconstruction. Only a few of the T cells detected in the Seurat reconstruction show *CD3D* and can therefore be considered as T cells (Figure 36C, right panel).

These T cells were visualized in the t-SNE representation, showing that most T cells, detected using both methods, show distinct clusters in the kidney-lq and the kidney-hq data (Figure 36D, right panel). However, T cells from the kidney-lq data set only show low expression of *CD3D* without reconstruction (Figure 36B, right panel). Interestingly, T cells which were detected only by DISCERN form a distinct kidney-lq-specific cluster also in the Seurat reconstructed data (Figure 36D, middle panel) indicating that the batch correction by Seurat is insufficient to integrate all available T cells. DISCERN improves the detection of T cells for this data set by providing a reconstruction, e.g. used for visualization and cluster detection, which is consistent with the observed expression in uncorrected data and thus easier to interpret.

Synopsis of Results In summary, DISCERN allowed for a faithful reconstruction of pathway-specific gene expression and a cell type-specific gene expression recovery for an snRNA-seq data set with corresponding scRNA-seq data. This analysis was performed in comparison with the established batch correction and imputation methods. The reconstruction by DISCERN shows the highest similarity to the reference scRNA-seq data and delivers a reconstruction of gene expression that is biologically reasonable. In a second experiment, the expression of selected marker genes in the pancreas data set was tested. Especially droplet-based sequencing technologies, e.g. InDrop, show more noisy expression patterns of the selected marker genes compared to well-based technologies, e.g. SMART-seq2. DISCERN can correct these technology-specific effects and improve marker gene detection in the pancreas data set.

The third experiment utilized an scRNA-seq reference to detect T cells in snRNA-seq data of patients with acute kidney injury. While the snRNA-seq data has a higher number of cells, only a few T cells could be detected. Reconstruction with DISCERN using the scRNA-seq data as a reference, resulted in increased marker



(A) t-SNE representation without reconstruction (left) and after reconstruction with Seurat (middle) and DISCERN (right). Coloring was done by batch.

(B) t-SNE representation without reconstruction (left) and after reconstruction with Seurat (middle) and DISCERN (right) colored by *CD3D* expression as a marker or T cells.

(C) Venn diagrams showing the overlap of detected T cells in kidney-lq and kidney-hq and their *CD3D* expression in uncorrected data.

(D) t-SNE representation without reconstruction (left) and after reconstruction with Seurat (middle) and DISCERN (right) colored by which method was able to detect these T cells.

Figure 36: T cell detection in kidney snRNA-seq (kidney-lq) and scRNA-seq (kidney-hq) data of patients with acute kidney injury. The data set, including the Seurat batch, reconstructed data was provided by Rajasree Menon from the group of Matthias Kretzler, University Michigan. The t-SNE representations were computed on the top 50 principal components. The T cell clusters were selected using Leiden clustering. For DISCERN reconstruction the kidney-hq was selected as the reference batch. Patient 3210003 was removed due to high batch effects and low quality of this patient.

gene expression and more complete consistent and simplified T cell detection, compared to Seurat. Thus, DISCERN improved marker gene detection across three experiments using another scRNA-seq data set as a reference.

3.3.2 Reconstruction using bulk RNA-seq

Detection of marker genes is challenging, due to high dropout rates and batch effects. However, DISCERN was shown to improve marker gene detection if high-quality data is available (Section 3.3.1). Thus, bulk RNA-seq data of purified cell types (e.g. FACS-sorted immune cells) could be a suitable hq proxy for the expected gene expressions per cell. To evaluate the ability of DISCERN to cope with bulk RNA-seq and scRNA-seq at the same time, the bulk data set [138], containing 9 852 FACS-sorted samples of different PBMC cell types, was used as a high-quality batch. The evaluation was done using the citeseq data set, where additionally the cells were labeled with 15 antibodies to improve cell type discovery using protein abundance information [137]. The CITE-seq information of nine antibodies, where reasonable amounts of protein abundance was present, was used to verify the expression reconstruction. This enabled the validation in cases, where the gene expression is absent, but protein abundance and cell type identity can be proven via antibody labeling (Figure 37).

This CITE-seq information (protein abundance) was not used in DISCERN training or reconstruction. Again, *CD3D* is used as a marker for T cells, i.e. CD4 and CD8 T cells, in particular. While *CD3D* is expressed almost exclusively in T cells, however, uncorrected cells show missing expression due to dropout. This effect can be reconstructed using DISCERN (Figure 37, first row). *CD4* is a marker for CD4 T cells, which is according to the protein abundance highly expressed in CD4 T cells and medium-highly expressed in Monocytes. Most of the cells showing CD4 protein abundance, do not show a corresponding gene expression before reconstruction, but show significant *CD4* expression after reconstruction using DISCERN (Figure 37, second row). Interestingly a small proportion of cell types, annotated and CD4 T cells, do not show CD4 protein abundance or expression of *CD4* before and after reconstruction by DISCERN. Instead, these cells show strong CD8 protein abundance, high expression of *CD8A* after reconstruction by DISCERN, and medium-high expression of *CD8A* before reconstruction (Figure 37, third row). In the original t-SNE representation, these cells do not show a distinct cluster (Figure 37, third row, first column) and were therefore most probably wrongly annotated as CD4 T cells. *CD2* is another marker for T cells and NK cells and shows high protein abundance in these cell types. This can also be observed in uncorrected data, but nearly all T cells and NK cells express *CD2* after reconstruction with DISCERN. For genes that are rarely expressed in the uncorrected data, for example *B3GAT1*, a marker for NK cells, DISCERN can reconstruct expression values (which were too low in the uncorrected data set) in cells with a high abundance of the B3GAT1 protein (Figure 37, fifth row). However, DISCERN is not able to fully reconstruct the *B3GAT1* expression in B cells as suggested by the protein abundance. Interestingly, the Human protein atlas also does not report *B3GAT1* expression in B cells ($0.0\,\mathrm{nTPM}$ in B cells and $20.7\,\mathrm{nTPM}$ in NK cells) [152].

Thus, it is not clear whether DISCERN correctly predicted that there is no expression of *B3GAT1* in B cells and the CITE-seq information is incorrect or whether *B3GAT1* is hard to detect in scRNA-seq based expression profiles of B cells.

CD14, *FCGR3A* and *ITGAX* are markers for Monocytes, which are detected by the protein abundance and reconstruction by DISCERN, but only partially from the uncorrected expression data (Figure 37, rows 6 to 8). *FCGR3A* is specific for a subpopulation of Monocytes but also expressed in NK cells. This is correctly captured by the DISCERN reconstruction (Figure 37, sixth row). *ITGAX* expression is found in Monocytes and NK cells in the DISCERN reconstruction while being nearly completely absent in the uncorrected data. The protein abundance shows ITGAX abundance only in Monocytes, however, also medium-high abundance is shown in NK cells. DISCERN is potentially over-estimating the *ITGAX* expression in NK cells when compared to the protein abundance (Figure 37, eights row). The expression of *CD19* is specific to B cells, as verified using the protein abundance. A few B cells expressing *CD19* are detected in uncorrected data (Figure 37, ninth row). DISCERN shows slight over-estimation of *CD19* expression in some NK cells and a few other cell types. However, the expression of *CD19* is generally very low compared to other marker genes, which makes robust imputation very difficult.

To investigate the ability of DISCERN to detect previously unseen cell types, the CD4 T cells from the citeseq-lq batch (see Figure 37) after reconstruction with bulk-hq data were selected and further analyzed. The detection of cell subtype-specific information for CD4^{+} T helper cells from PBMC scRNA-seq data is difficult, since they show a limited activation status in healthy individuals. However, they are

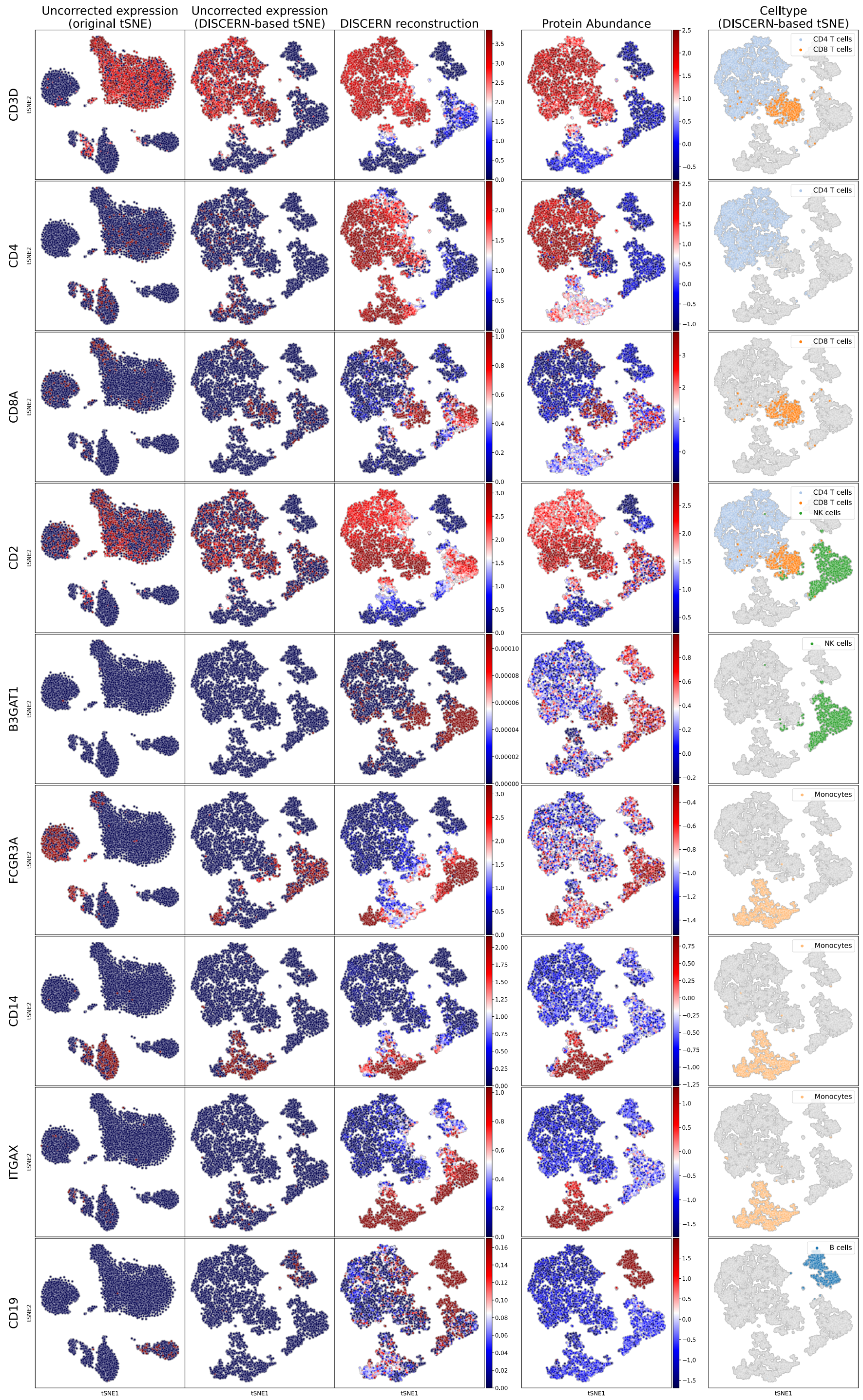


Figure 37: t-SNE representation of the uncorrected and gene expression reconstructed by DISCERN as well as corresponding protein abundance for the citeseq dataset. Expression levels are displayed without reconstruction (first and second column) and after reconstruction using the bulk-hq data set as reference (third column). Protein abundance measured using CITE-seq is shown in the fourth column. Cell types, where the expression of the gene is expected, are shown in the fifth column. The first column shows the t-SNE representation computed for the uncorrected data, while the others are computed on the reconstructed expression data. For the uncorrected and the reconstructed expression (first three columns), the same color scale is displayed. The protein abundance (CITE-seq) is mean-variance scaled.

commonly detected using FACS after stimulation indicating their existence in healthy individuals [153]. Clustering with the Leiden algorithm [141] using highly variable genes of citeseq-lq data resulted in the detection of TH17, TH2, TH1, HLA-DR expressing TREG (Active_TREG), naive CD4$^+$ T cells (CD4_naive), effector-memory CD4$^+$ T cells (CD4_EM), central-memory CD4$^+$ T cells (CD4_CM), and effector cells expressing IFN-regulated genes (IFN_regulated) (Figure 38, second panel). For uncorrected data an obvious clustering cannot be observed. However, clusters found in DISCERN-reconstructed data can, in most cases, visually be found in uncorrected data as well, e.g. IFN_regulated and TH1 (Figure 38, first panel).

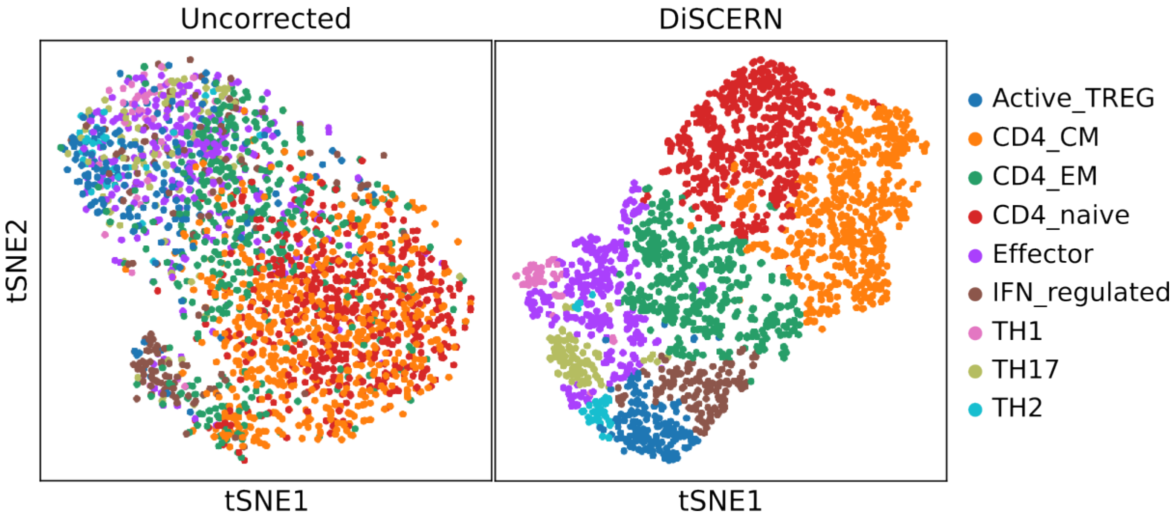


Figure 38: t-SNE representation of CD4$^+$ T cells in the citeseq dataset after annotation of the cell types found after expression reconstruction with DISCERN. The first panel shows the t-SNE computed on uncorrected data and the second panel shows the t-SNE computed on the DISCERN reconstructed data. CD4$^+$ T cells were selected as CD4 T cells in Figure 37 excluding cells, which show a CD4 protein abundance lower than 2.5 and an expression value of *CD3E* < 2.5. Highly variable gene selection was performed using the protocol implemented in Seurat v3 and the top $1\,000$ genes of high variability were used for computing the PCA. Cell type annotation was provided by Can Ergen-Behr [154].

To verify the plausibility of the detected cell subtypes, the expression of marker genes before and after reconstruction was investigated (Figure 39). Unfortunately, for most of the markers, no expression could be detected in the uncorrected data. However, after reconstruction all detected T cell subtypes show corresponding marker gene expression, but it can also be observed that the marker gene expression is often not exclusively present for a specific cell type. This is a common problem for T cell subtype identification and remains a challenge in scRNA-seq data analysis of T cells [155].

To verify that these cell types are not derived from the bulk-hq reference, the proportions of the different cell types were calculated and compared to data from the literature (Figure 40). Since the bulk-hq data set was constructed to contain several samples of T cell subtypes, it does not represent the real proportions of these cells in healthy PBMCs.

Applying DISCERN to reconstruct expression levels following by subsequent cluster annotation, reveals that the estimated cell-type proportions reflect the proportions derived from the literature more closely. This indicates they are not adopted from the bulk-hq data but from their real biological state.

In summary, DISCERN can utilize the comprehensive expression information available through bulk RNA-seq data for the reconstruction of scRNA-seq data. The reconstructed expression values were verified by protein abundance information coming from CITE-seq. Furthermore, the quality-improved data set could be used to distinguish CD4$^+$ T cell subtypes, which are often only detectable after stimulation. The presence of these cell subtypes was verified using marker gene expression in reconstructed and uncorrected expression data. Finally, the proportions of selected subtypes were compared to literature-based and FACS-based studies, showing a similar abundance as detected after DISCERN reconstruction.

3.3.3 Case study: COVID-19

In the previous section, DISCERN-based reconstruction using bulk RNA-seq reference was shown to be beneficial for the detection of cell subtypes. Therefore, DISCERN was used to investigate the cell type-composition in COVID-19 using a previously published data set [140] consisting of lung and blood immune cell scRNA-seq data. Again the bulk-hq [138] data set was used as a reference for the reconstruction of expression values by DISCERN.

The COVID-19 blood data set (covid-blood-lq) was originally analyzed using Seurat [140], but only a limited resolution of T cell subtypes could be identified. Although CD4$^+$, CD8$^+$, and NK cells could be detected, the UMAP representation did not allow to distinguish subpopulations of these cells in covid-blood-lq data [140].

However, reconstruction by DISCERN enabled the identification of 24 subtypes of CD4$^+$ and CD8$^+$ T cells in covid-blood-hq data. The cell type annotation was pro-

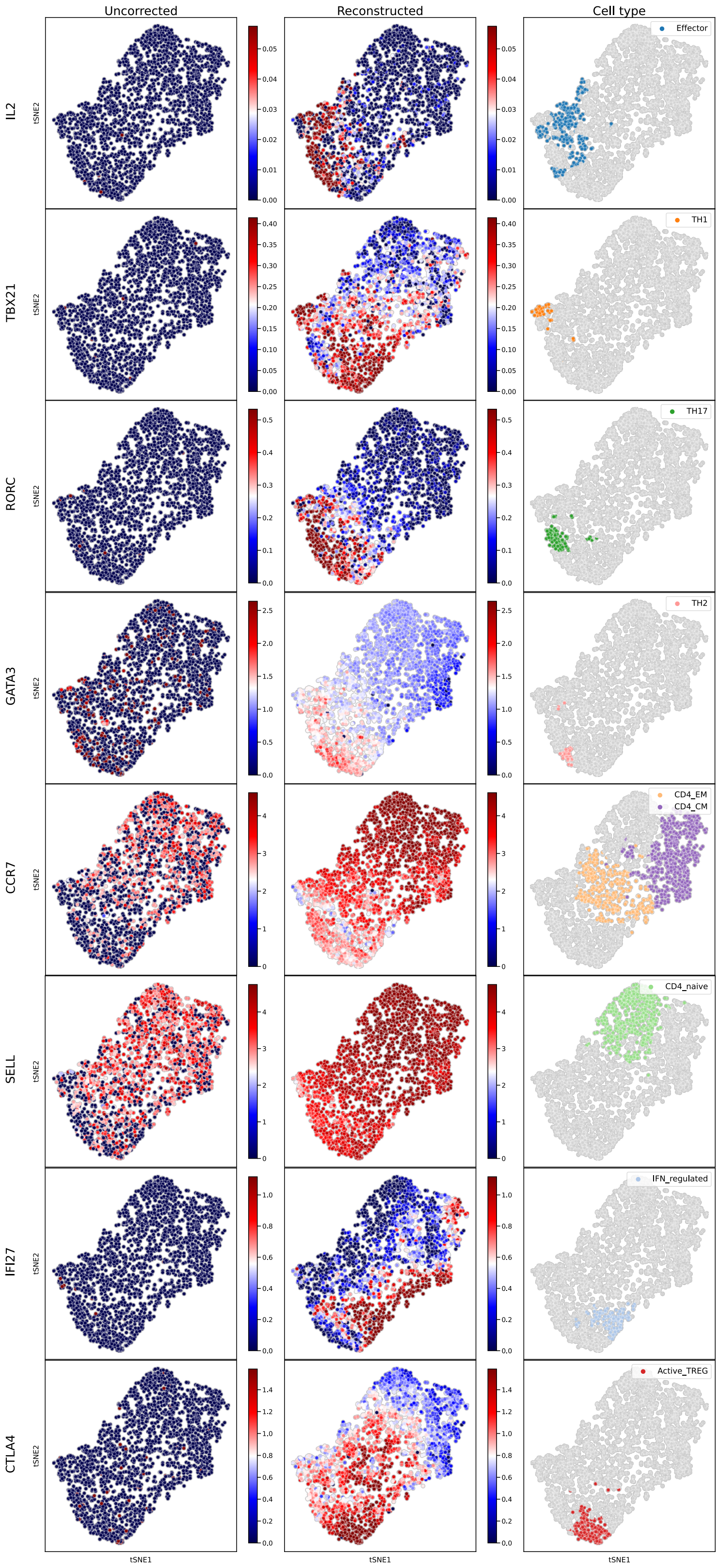


Figure 39: t-SNE representation of CD4^{+} T cells in the citeseq data set and cell type marker genes. The first column shows the expression values of the marker genes in uncorrected data and the second column shows expression levels in data reconstructed by DISCERN. The third column shows the cell type which is expected to express the marker gene.

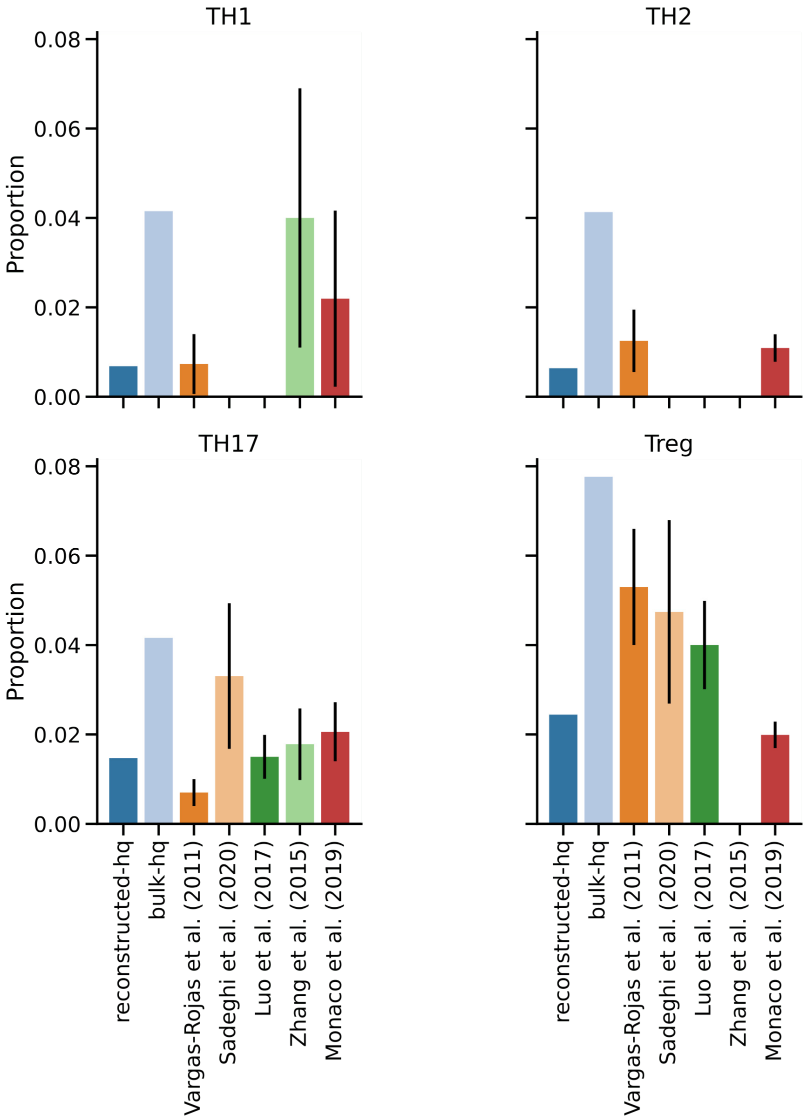


Figure 40: Proportions of CD4$^+$ T helper cell subtypes (TH1, TH2, TH17, and Treg) identified in the citeseq-lq data after DISCERN-based reconstruction using bulk-hq as a reference, and published ground truth cell fractions in PBMC data. The proportions are calculated concerning the total number of PBMCs. To compare the proportions with existing literature, five studies were considered [153, 156–159]. These studies estimated one or more of these subtypes using FACS and subsequent cell stimulation. For these studies, the error bars represent one standard deviation. Missing bars indicate that the corresponding cell type was not quantified.

vided by Can Ergen-Behr [154].

Interestingly, TH17 cells occur in two separate clusters, called TH17_cluster1 and TH17_cluster2 (Figure 41). The expression of the TH17 marker genes *IL17A* and *IL17F* is very low (mean expression of 4.0×10^{-5} for *IL17A* and 7.2×10^{-5} for *IL17F* in TH17 cells) in the uncorrected data (covid-blood-lq), but increased approximately by a factor of 10 in the expression data reconstructed by DISCERN (covid-blood-hq). In covid-blood-hq the mean expression was 5.9×10^{-4} for *IL17A* and 5.6×10^{-4} for *IL17F* in TH17 cells after reconstruction. Both clusters show, with and without reconstruction, expression of *RORC* (mean expression of 0.2 with and 0.02 without reconstruction), which is known to be important for the differentiation of TH17 cells[160].

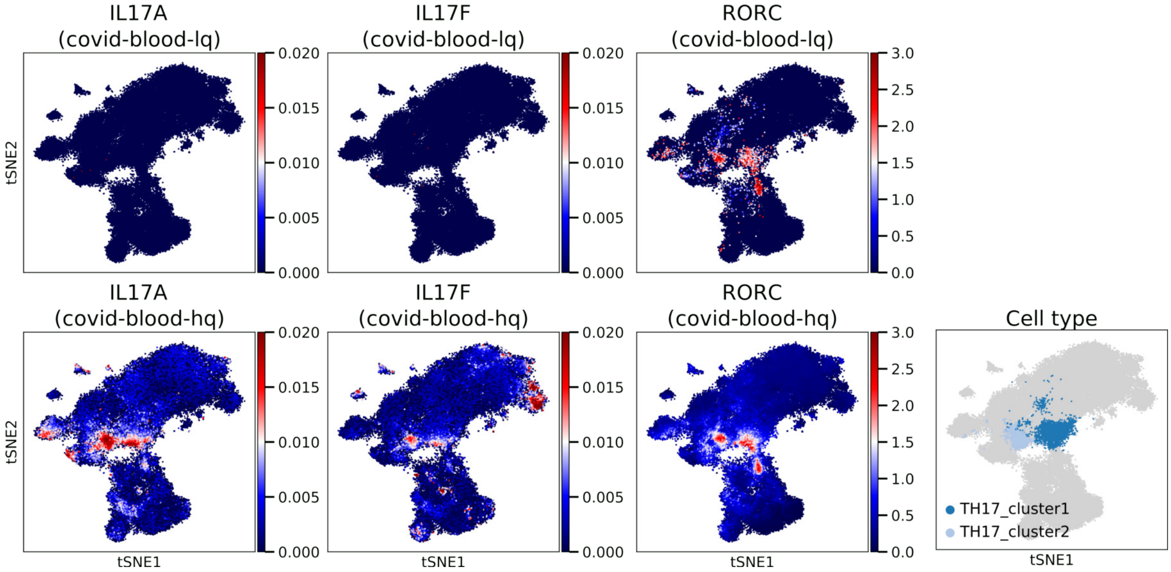


Figure 41: t-SNE representation of TH17 marker genes in two TH17 subtypes detected in COVID-19 patient blood. t-SNEs were calculated for CD4^{+} T cells on covid-blood-hq data. The first row shows the expression of marker genes for uncorrected covid-blood-lq data. The second row displays the expression of the same marker genes for reconstructed covid-blood-hq data. The covid-blood-lq data was reconstructed using the bulk-hq reference to obtain covid-blood-hq data. The TH17 cell subclusters were found by Louvain [161] clustering after reconstruction. Colors represent the expression levels of genes as mentioned in the plot titles (*IL17A*, *IL17F*, *RORC*; from left to right). Expression levels of TH17 marker gene expression are barely visible for *IL17A/F* before reconstruction but can be detected after reconstruction with DISCERN. *RORC*, as transcription factor for TH17 cells, confirms the correct annotation of TH17 cells.

Considering major T cell activation markers, *HLA-DRA*, *HLA-DRB1*, *CCR4* and *RBPJ*, cluster 2 can be annotated as activated TH17 cells, whereas cluster 1 shows minor activation markers and thus exhibits a memory T cell phenotype. The expression of *RBPJ* is of further interest because it was shown to be linked to TH17 cell pathogenicity [162]. This may suggest a role of pathogenic TH17 cells in COVID-19 and bacterial pneumonia. A similar TH17 phenotype was observed in COVID-19 after stimulation of T cells [163].

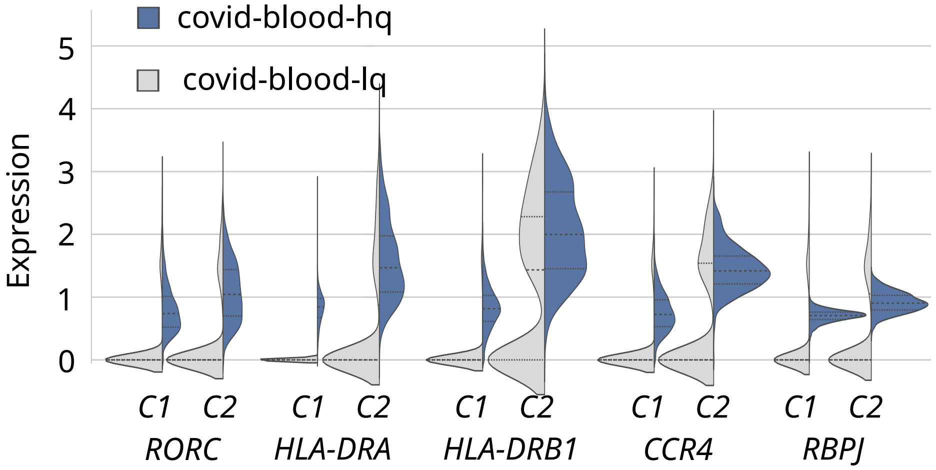


Figure 42: Violin plots showing expression levels for genes distinguishing TH17_cluster1 (C1) and TH17_cluster2 (C2) cells without (covid-blood-lq) and with (covid-blood-hq) reconstruction using DISCERN and the bulk-hq reference. *RORC* is known to be important for TH17 differentiation [160], *HLA-DRA*, *HLA-DRB1*, *CCR4* and *RBPJ* are known activation markers for T cells [162, 164–166]. Violin plots indicate expression for uncorrected data (covid-blood-lq) and after reconstruction using DISCERN (covid-blood-hq).

To further validate the presence of this cell type, without considering data reconstructed by DISCERN, the COVID-19 lung data set (covid-lung) was used. The covid-blood and the covid-lung data where sequenced using TCR-seq. TCR-seq is a method to assess the RNA sequence of the TCR in each T cell during scRNA-seq [167]. The DNA sequence of the TCR gets re-assembled during maturation from hematopoietic stem cells, such that each T cell can uniquely be identified by the TCR sequence (RNA and DNA). The sequence of the TCR additionally determines by which epitope(s), i.e. by which peptides presented by antigen-presenting cells, a T cell is activated. During activation, a T cell divides and forms multiple activated clones sharing the same TCR [167]. As a consequence, by TCR-seq, for each T cell their clonotype (unique TCR sequence) can be determined and used to trace cells in multiple tissues.

Thus, the clonotype was used to determine for each cell its corresponding cell type annotated in the covid-lung data set. Since cells with the same TCR in lung and blood originate from the same progenitor, they have a high probability of belonging to the same cell type. In the covid-lung data, memory cells, T cells and TH17 cells were readily observed without reconstruction [140].

The clonotype of TH17_cluster1 cells is frequently overlapping with CD4$^+$ memory T cells in covid-lung (Figure 43, CD4_TCM). In comparison, the TH17_cluster2 in blood shows strong clonal overlap with effector memory and resident memory TH17 cells in covid-lung data (Figure 43, TEM17 and TRM17). These findings support the hypothesis that both cell types exist in this data set and have a similar function in blood and in the lung.

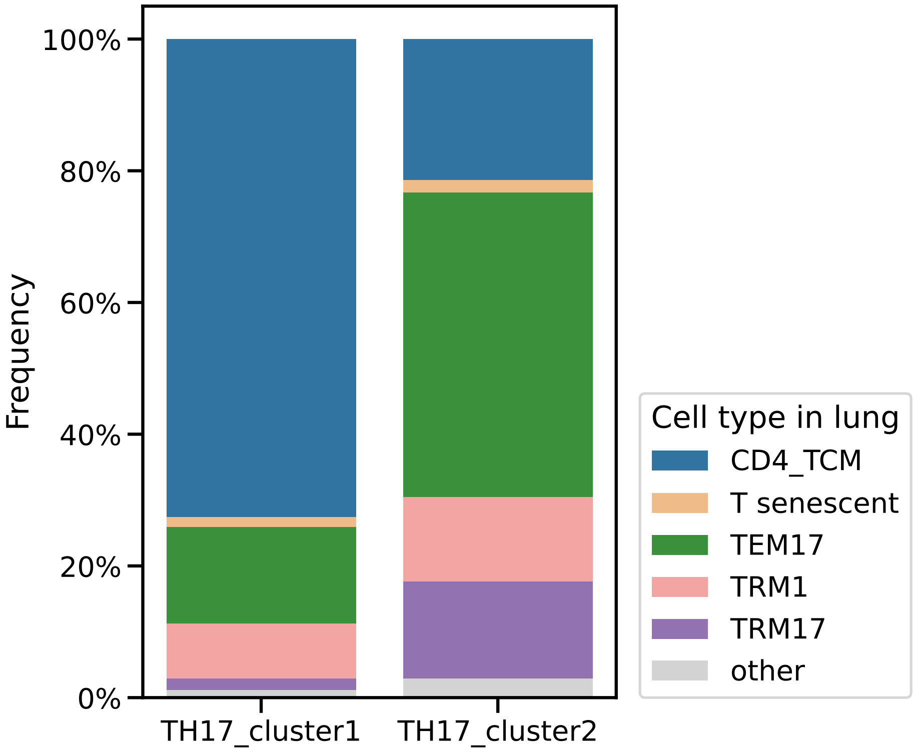


Figure 43: Fraction of TH17 cells sharing the TCR clonotype in covid-blood-hq and covid-lung data. Cell-type annotations of lung data were used as provided in the original publication [140]. Cell types with an overlap of less than 1 % in both TH17 clusters were labeled as other. TH17_cluster1, detected in covid-blood-hq data, shares TCR clones with CD4_TCM cells in the covid-lung data. TH17_cluster2, detected in covid-blood-hq data, shares most T cell receptor clones with TEM17 cells in covid-lung data. This corroborates the correctness of the nomenclature of the two TH17 subtypes detected in covid-blood-hq data.

To verify that the shared cells in the lung also exhibit a TH17-like phenotype, the expression levels of TH17 markers, *RORC* and *IL17A*, were investigated. Cells sharing their clonotype with TH17_cluster1 and TH17_cluster2 cells in the blood, express *RORC* slightly lower than TEM17 cells, but higher than CD4$^+$ memory T cells (CD4_TCM) on average (Figure 44). *IL17A* is highly expressed in cells, which share their clonotypes with TH17_cluster2, and lowly expressed in cells, which share their clonotype with the TH17_cluster1, similar to TEM17 cells. CD4_TCM show nearly no expression of *IL17A* suggesting, that cells sharing clonotype with TH17 clusters are indeed TH17 cells (Figure 44). Cells sharing their clonotype with TH17_cluster2 show a higher expression of *RORC* and *IL17A*, indicating a stronger activation pattern than cells sharing their clonotype with TH17_cluster1 (Figure 44).

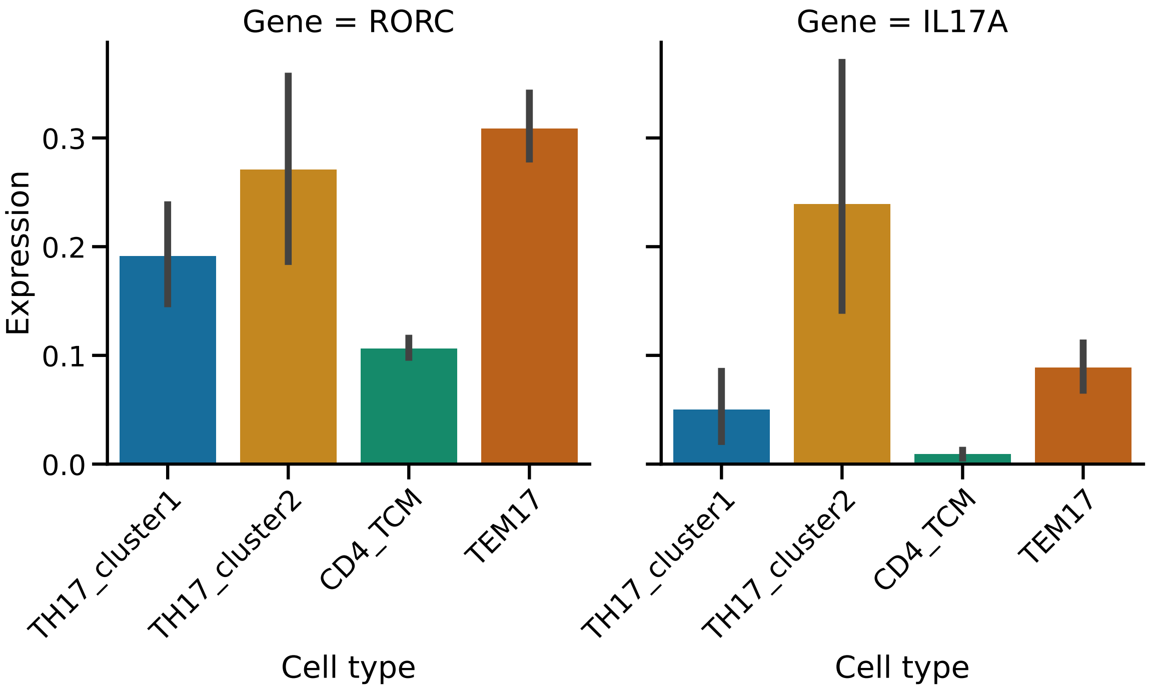


Figure 44: Mean expression of *RORC* and *IL17-A* of covid-lung cells sharing a clonotype with TH17 cells of the covid-blood data. TH17_cluster1 and TH17_cluster2 are determined using the TCR clonotype information of reconstructed covid-blood-hq data and CD4_TCM or TEM17 covid-lung cells were annotated as in the original publication [140] (see also Figure 43). A single cell can contribute to more than one bar, e.g. by being annotated as TEM and having a shared clonotype with TH17_cluster2 cell in covid-blood. Cell types sharing a clonotype with TH17_cluster1 and TH17_cluster2 cells from covid-blood have on average a higher or similar expression of the TH17 marker genes (*RORC* and *IL17A*) than cells in CD4_TCM or TEM17 cells in the lung.

To investigate the effect of TH17_cluster1 (C1) and TH17_cluster2 (C2) cells on bacterial pneumonia and COVID-19, the ratio of these cell types was compared (Figure 45). The number of patients is too small, especially because there are only three patients with bacterial pneumonia that have these cell types, to conclude that the observed shift in proportions of cell types in C1 and C2 is statistically significant. Thus, it remains unclear if this shift in proportions is specific to COVID-19 or if it is generally observed in severe lung damage.

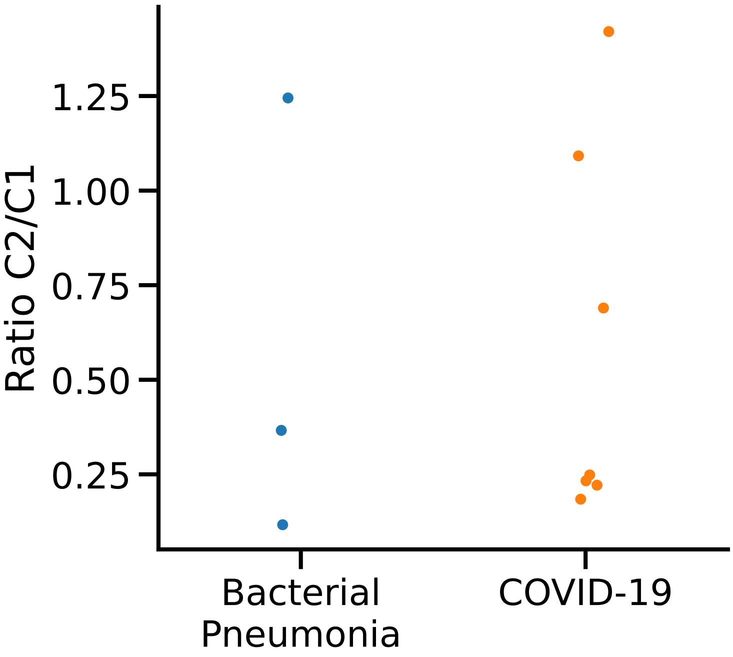


Figure 45: Ratio of cell types annotated as TH17_cluster1 (C1) and TH17_cluster2 (C2) in covid-blood for each patient. The colors indicate the disease of the patient, either bacterial pneumonia or COVID-19. The data set consists of three patients with bacterial pneumonia and seven patients with COVID-19. The ratio was computed on the total number of cells in C1 and C2 respectively.

In summary, DISCERN enabled the detection of two activation states of TH17 cells in COVID-19 without stimulation. This was previously only possible using stimulation. This *in silico* approach has the advantage, that it does not add additional cost compared to *in vitro* stimulations and can be applied to already published data sets as well.

The findings could be verified using marker genes in uncorrected data and data reconstructed by DISCERN. Furthermore, independent TCR-seq information, not considered by DISCERN, provided a means to verify the presence of two subtypes of TH17 cells in covid-lung data. In combination with previous finding, this information suggests a potential influence of these newly detected subtypes in the development and severity of lung diseases.

4 Discussion

4.1 Common obstacles in expression reconstruction

Common scRNA-seq analyses, for example cell clustering and cell type identification, are impaired by the sparsity of gene expression information and the high level of technical noise (e.g. batch effects). Common ways to address these problems are batch effect correction and the imputation of missing gene expression information. To address the imputation problem, several algorithms such as scImpute, MAGIC, DeepImpute, and DCA have been developed. They impute missing gene expression in scRNA-seq data by utilizing expression information from similar cells within the same batch. Gene imputation can clearly improve gene expression by inferring values that were previously absent, due to technical reasons depending on the sequencing method. However, Andrews & Hemberg [115] showed that several imputation algorithms, even those reaching state-of-the-art performance, increase the number of false positives by imputing expression value for genes, which do not have a corresponding RNA expressed [115]. There are various reasons for biologically absent expression, e.g. transcriptional repression or mutations. Additionally, Ly & Vingron [168] found that imputation can decrease the performance of downstream applications, i.e. gene network inference [168], by introducing false positive gene-gene correlations.

While these imputation methods generally reduce the sparsity of expression data, they often violate the statistical properties assumed by downstream algorithms [115], e.g. a negative binomial distribution for modeling the measured expression. Common methods for downstream analysis, however, rely on these properties of scRNA-seq data. Therefore new methods need to be developed and existing methods need adaptation to the statistical properties of imputed scRNA-seq data.

4.2 Deep single-cell expression reconstruction

To solve the described problems, DISCERN was developed. The approach was to use a reference batch as additional information and impute realistic expression values in the sense that statistical properties of the input data are retained. As a consequence, established downstream analysis of the resulting data can be used without modifications.

DISCERN is a Wasserstein autoencoder (WAE)-based method, following the concepts proposed in Tolstikhin *et al.* [91] and adapted for expression modeling with multiple batches. This is achieved by conditional instance normalization (CIN) in the encoder and decoder and a dropout modeling layer in the decoder. The use of CIN enables DISCERN to flexibly choose a high-quality reference batch e.g. a specific batch showing an enriched expression of specific genes or a batch, which uses a different RNA isolation protocol. A reference batch can be chosen after model training, so that the reference batch can be varied depending on the scientific question to be solved.

In cases where it is not obvious which reference batch to choose, the ability of DISCERN to use different "high quality" batches, yielding multiple reconstructions, can be of advantage. Additionally, the dropout estimation procedure in the decoder of DISCERN yields expression values whose distribution is very similar to an scRNA-seq-data set. That is, the statistical assumptions about scRNA-seq data still hold for the output of the decoder and so DISCERN is compatible with common downstream applications. Thus, DISCERN can be used for expression reconstruction, including batch correction based on a reference data set. This can be any expression data set, e.g. scRNA-seq or bulk RNA-seq.

Usually, deep learning methods require extensive optimization of hyperparameters [130]. DISCERN showed robustness to the choice of hyperparameters and almost no difference in the performance of reconstruction of mean expression was observed for a wide range of hyperparameters, when applied to the pancreas data set (Figure 9). The stability of DISCERNs hyperparameters suggests that no extensive hyperparameter optimization is required. This simplifies its application and saves considerable resources in terms of time and energy.

4.3 Benchmark of expression reconstruction methods

To evaluate the performance of DISCERN and competing tools, extensive benchmarks of batch correction, of imputation using real data, and of imputation using *in silico* modified data were performed. To mimic a realistic use case scenario, all methods, including DISCERN, were applied using their default settings.

Due to missing expression information and batch effects of multiple sequencing experiments, the evaluation of correct imputation and batch correction is challenging. In batch correction evaluation a model is considered the better, the more it can preserve existing biological features and integrate different batches. Since it is often not possible to reliably distinguish batch effects from biological effects, metrics measuring these terms are usually contradictory to each other.

Varying numbers of cell types or cell numbers per batch are particularly challenging for batch correction tools and their evaluation. For example, cells from a cell type, that is exclusively present in one batch, should not be integrated with cell types from other batches. Also, evaluation of imputation methods is challenging, because usually no ground truth information about the expression values is available. Therefore, the expression values are compared to expression values from other experiments or to held-out data. This ground truth information is limited due to batch effects. To overcome the problem of missing ground truth information in imputation experiments, simulation methods have been developed. These methods usually rely on certain assumptions about scRNA-seq data. For example, Splatter [169] uses a gamma-Poisson distribution, and thus favors models, which are good in reproducing these assumptions without necessarily working well on real experimental data. Therefore, expression reconstruction, including imputation and batch correction, was not evaluated on simulated data sets, but mainly on unmodified data sets. These consist of data from multiple sequencing technologies (pancreas), RNA isolation techniques (snRNA-seq & scRNA-seq), and various batches (difftec) to cover a wide range of conditions occurring in common sequencing experiments.

To measure imputation performance, all experiments were evaluated using measured expression information without relying on simulation tools. The experiments either used a reference batch as an evaluation criterion or held-out expression information. Furthermore, to capture differences between experiments, the effect of reference batches of different sizes and the effect of overlap of cell types was evaluated. An overview of the best-performing methods in all benchmarks can be found in Table 9.

In the batch correction benchmark using adjusted Rand index (ARI), the Silhouette score and adjusted mutual information (AMI) on the data sets pancreas, sn/sc, and difftec, DISCERN, scVI, and Seurat were the top performing methods (Table 4). Seurat was explicitly developed for batch correction. DISCERN and scVI achieve batch correction performance by model architecture. While DISCERN includes the batch information using CIN, scVI introduces the batch information as an additional input to each layer in the neural network.

In the gene expression imputation experiments evaluated, DISCERN, scGEN, and scVI achieve the best performance, when using mean expression, variation calculation, dropout estimation, and differential expression analysis (DEA) (Table 5) as ranking criterion. In these evaluations, the three best-performing methods all use a reference batch. This suggests, that the use of a reference batch is beneficial for imputation.

In the experiments involving *in silico* gene removal, DISCERN, scVI and CarDEC achieve the best performance (Table 6). Unlike DISCERN and scVI, CarDEC does not use a reference batch. However, the relatively large rank-sum of 29.0 for CarDEC indicates a gap between the first two and the third best model. DISCERN achieves a rank sum of 15.0 and scVI of 16.0.

In the experiments using varying ratios of high- and low-quality batches and cell type overlaps, scVI, DISCERN, and CarDEC achieve the best performance (Table 7). scVI showed good performance in the reconstruction of $\log_2$-fold-changes, whereas DISCERN and CarDEC showed excellent performance in the reconstruction evaluated by t-statistics.

Overall (Tables 8 and 9) DISCERN, scVI, and scGEN are the best-performing methods across all benchmarks. These methods support the use of a reference batch for projection, which is advantageous for the conducted imputation, gene removal, and varying ratio experiments, but does not affect the batch correction evaluation, which is independent of the reference batch. Especially scVI and DISCERN show very similar performance in several conducted benchmarks. Compared to scVI, DISCERN has the advantage of allowing multiple batches as a reference, a feature described in Section 3.2. scVI is restricted to use a single batch input. Furthermore,

Table 9: Overview of the three best performing methods across all quantitative benchmarks (Table 8).

	Batch correction (Table 4)	Imputation (Table 5)	*In silico* gene removal (Table 6)	Batch & cell type ratio (Table 7)
First rank	DISCERN	DISCERN	DISCERN	scVI
Second rank	scVI	scGEN	scVI	DISCERN
Third rank	Seurat	scVI	CarDEC	CarDEC

DISCERN is the best-performing method for nearly all DEAs considering statistical significance compared to the $\log 2$-fold-change. In contrast, scVI achieves usually the best performance in all $\log 2$-fold-change experiments, but only second place for the evaluations considering statistical significance. This indicates, that scVI can model the general tendency of the gene expression values, but is not able to maintain the expected distribution. This can potentially lead to an increased false-positive rate since statistical significance is often used to select deregulated genes for further analysis [66]. Therefore, better modeling of significance, as in DISCERN, is favorable.

4.4 The network architecture

While scGEN and scVI are based on the variational autoencoder (VAE) architecture, DISCERN uses the WAE architecture. VAEs have a similar architecture as WAEs (see Sections 1.4.3 and 1.4.4 for details), but differ in the comparison of the prior distribution. WAE allow the use of several loss functions in contrast to VAEs, which require the KL-Divergence. Whether the use of different loss functions affects the model performance, is hard to determine, but the flexibility in choosing the loss function makes the development of networks more flexible.

Additionally, WAEs evaluate the difference between the prior distribution and the aggregated posterior distribution, contrary to the posterior distribution in VAEs. This has the advantage of better modeling samples that are very different, providing a smoother embedding and reduction of blurriness in generated data [91, 104, 170]. Multiple extensions of WAE networks have been developed and tested for the reconstruction of image data (e.g. MNIST [171], CELEB-A [172]). These extensions were proposed to have better metrics for comparing the embedding to the prior distribution [170, 173]. However, a deeper evaluation of WAEs for more advanced data sets is still missing. Hence, WAEs are not frequently applied and VAEs remain the most widely used architectures for the analysis of single-cell data sets [174, 175].

4.5 Applications of DISCERN

The comprehensive benchmark for batch correction and imputation conducted in this work is the foundation for an application-oriented evaluation of DISCERN to gain new knowledge on biological processes based on scRNA-seq data. Further evaluation of DISCERN was done using the snRNA-seq/scRNA-seq data set, the pancreas data set and the kidney data set.

DISCERN showed improved reconstruction of cell type-specific pathway expression in the sn-lq (snRNA-seq) batch. This was more similar to the sc-hq (scRNA-seq) batch, which was considered as ground truth for the evaluation, in comparison to all other methods tested. Qualitatively, the similarity can easily be discovered visually (Figures 32 and 33), but the conducted min-max scaling for visualization has an effect, that can easily be underestimated and therefore obscure the similarity of gene expression. Thus, the dissimilarity can potentially be explained by slight variations in the minimum and maximum in the case of min-max scaling or the mean and variance in the case of mean-variance scaling. However, a high similarity of the gene expression indicates that the expression values are close to the expected values. Hence, DISCERN achieves realistic expression values without having scaling issues on the considered data set.

Pairwise similar genes, like *CD3D* and *CD3E*, are highly expressed in T cells of the scRNA-seq/snRNA-seq data set. Interestingly, some genes, e.g. *CD3G*, for which high expression is expected, show low expression before reconstruction, after reconstruction, and in the high-quality reference data (Figure 32A). This exemplifies that some genes, where high expression values are expected, are not detected in both, scRNA-seq and snRNA-seq data. In these cases, it is not possible to reconstruct the expression values with DISCERN. In contrast, other tools reconstruct high expression values (e.g. CarDEC in Figure 32A), even though a gene, e.g. *CD3G*, was potentially not expressed and thus is not biologically valid.

This makes the evaluation of reconstructed expression values using prior knowledge challenging, since not every data set adequately represents all aspects of prior knowledge. The direct comparison to a related data set, e.g. coming from the same sample and tissue, can serve as a reference for comparison and gives a more realistic overview of the cellular changes and expressed marker genes.

To investigate the ability of DISCERN to capture cell type-specific marker gene expression and integrate them across different sequencing technologies, the pancreas data set was used (Figure 34). Here, a beta-cell-specific marker Insulin (*INS*), an acinar-cell-specific marker Regenerating Family Member 1 Alpha (*REG1A*), and a delta-cell-specific marker Somatostatin (*SST*) were selected, based on results from the literature [147–149]. For all markers, the corresponding cell types indeed show high marker gene expression across all sequencing technologies.

However, the low-quality batches, indrop and celseq, show a diffuse expression pattern across all cell types. This makes the detection of cell type-specific expression difficult, because the batch effect overshadows the expression pattern. DISCERN can remove these unexpected expression patterns by reconstruction using the high-quality batch (Figures 34A to 34C, middle column). However, DISCERN is also able to apply the unexpected expression pattern in all batches, by projection to the low-quality batch (Figures 34A and 34C, last column). Thus, the selection of the reference batch needs to be applied carefully, even if the impact on the clustering is marginal (Section 3.2 and Figure 31). The number of expressed genes and the average gene expression per cell can give a reasonable approximation to assess the data set quality, but there are cases where these metrics can be misleading. For example, in droplet-based sequencing technologies, high proportions of ambient RNA can lead to low expression of cell type-specific genes in cell populations, which should not express these genes [176]. Ambient RNA is RNA, which does not originate from the sequenced cell, but from other cells destroyed in the cell extraction procedure before sequencing. This RNA is incorporated in droplets with cells by chance and introduces low values of measured gene expression for not expressed genes [176]. This could explain the unexpected expression pattern of *INS* in the indrop batch (Figure 34A). This artificially increases the number of expressed genes of actual low-quality batches.

While autoencoders are suitable for denoising ([84, 177]), the application of ambient RNA removal tools, e.g. SoupX [178] or DecontX [176], before expression reconstruction methods remain largely unexplored. Thus, the use of DISCERN for ambient RNA removal could be investigated in further work in addition to the application of ambient RNA removal tools before employing DISCERN.

Another effect observed in the reconstruction delivered by DISCERN is the unexpected high expression of *REG1A* in macrophages of the celseq batch (Figure 34B). When investigated in detail, this high expression is based on a single cell annotated as “macrophage”. This cell locates closely in the t-SNE representation with acinar cells when the celseq batch is analyzed alone (Figure 35). It seems that this cell could also be a wrongly annotated acinar cell. However, the annotation of a single cell in scRNA-seq data is difficult, due to the high dropout rate. Thus the exact cell type of this cell should be considered unknown, and it is important to note that in the reconstruction by DISCERN the high expression level of *REG1A* in this cell is retained. In contrast, expression reconstruction tools utilizing the cell type annotation could remove the expression of *REG1A*, which is unexpected for macrophages, in this cell.

While the choice of sequencing technology plays an important role, also RNA extraction can impact the experimental outcome. As seen in Figures 32 and 33, the gene expression can be different between snRNA-seq and scRNA-seq. Furthermore, batch correction is less successful for data in which snRNA-seq and scRNA-seq is combined (Figure 15) compared to scRNA-seq-only data sets (difftec, pancreas). This makes the reconstruction of snRNA-seq data sets using scRNA-seq data sets challenging, as was shown for the kidney data set (Figure 36C), in which the number of T cells differs considerably between snRNA-seq (kidney-lq) and scRNA-seq (kidney-hq) data. Data reconstructed by DISCERN can be used to robustly identify T cells in both snRNA-seq and scRNA-seq data. These T cells do not form a distinct cluster in data reconstructed by Seurat (Figure 36D). While DISCERN does not achieve perfect integration of snRNA-seq and scRNA-seq data, it improves on uncorrected data and on data reconstructed by Seurat (Figure 36A). It also enhances and improves the T cell detection when using *CD3D* as a marker gene. The reconstruction of gene expression is largely limited by the completeness of the expression values of the high-quality batch, which itself is prone to dropout. The kidney data has some interesting features which makes it very challenging. Thus the evaluation of the qualitative performance of other expression reconstruction tools (besides Seurat and DISCERN) for this data set, could be an interesting topic of future research.

In contrast to scRNA-seq, bulk RNA-seq of sorted cell types provides a reference which is almost free of dropouts and thus fits well for the task of expression reconstruction. The use of bulk RNA-seq of sorted immune cells [138] in combination with the citeseq data set allowed DISCERN to detect new cell subtypes in peripheral blood mononuclear cells (PBMCs) (Figure 38). Furthermore, the use of the citeseq data set allowed for the assessment of the reconstruction performance after incorporating another data modality, namely protein abundance obtained by CITE-seq (Figure 37). However, when each cell is considered individually, the correlation between protein abundance and scRNA-seq is rather low (Pearson correlation of 0.04 to 0.53 [137], depending on the gene). Therefore, this evaluation was only done qualitatively. DISCERN achieved good performance of CITE-seq and reconstructed scRNA-seq expression values for all considered genes (Figure 37). While the clustering of uncorrected data already allowed to differentiate cell types, no obvious grouping of these cell types was visible. After clustering of data reconstructed by DISCERN cell types form separable groups (Figure 38). However, the expression values of nearly all markers were absent in the uncorrected data (Figure 39). This makes verification based on measured expression values of these subtypes in reconstruction delivered by DISCERN challenging. The fact that DISCERN is not hallucinating gene expression values, as verified using CITE-seq (Figure 37), increases the confidence in the correctness of the cell type annotation. Hence, reconstruction is necessary to annotate these cell types in this data set.

Several of these cell subtypes require stimulation [179] to form well-defined clusters in PBMCs. Theoretically, these subtypes could result from the reconstruction using the sorted bulk RNA-seq. Thus, the proportions of the detected cell types were compared using Fluorescence Activated Cell Sorting (FACS) (Figure 40). This evaluation shows that the cell type proportions could reflect real proportions, but is not sufficient to prove the presence or absence of these cell types.

Because the proportions of cells in the sorted bulk RNA-seq data set hardly show consistencies with the proportions of cells published in the literature, the bulk data set does not reflect the composition of PBMCs. However, such consistencies are achieved after the reconstruction by DISCERN in the citeseq data set. This indicates that there are no hallucination effects introduced by the use of bulk RNA-seq.

For the identification of CD4$^+$ T cell subtypes in scRNA-seq data, Ding *et al.* [179] recommend high-throughput droplet-based methods, e.g. 10x Chromium. But these authors also found that good-quality data sets can improve the detection of these cell subtypes. To this end, DISCERN offers the possibility to incorporate low-quality data from high throughput methods and to improve its quality by using low-throughput well-based scRNA-seq or bulk RNA-seq data sets as a reference [180].

4.6 COVID-19 case study

It was already shown that the use of the bulk RNA-seq data sets improves the cell subtype detection in the citeseq data set. So it seems possible that in the future high-quality data sets, e.g. from stimulated T cells, could be used to improve the quality of data from droplet-based sequencing experiments.

Since reconstruction by DISCERN based on a bulk RNA-seq reference improved T cell subtype detection in PBMCs, it seems reasonable to apply the same approach to a coronavirus disease 2019 (COVID-19) data set of PBMCs, collected from patients who suffer from COVID-19 or pneumonia caused by bacterial infections. The reconstruction by DISCERN of the COVID-19 blood PBMC data using the bulk RNA-seq data led to the detection of two TH17 subclusters, with distinct activation patterns (Figures 41 and 42). These clusters show strong clonotype overlap with T cells analyzed in lung, i.e. bronchoalveolar lavage (BAL), of the same patients. This finding supports the annotation of cells in TH17_cluster1 as naive TH17 cells and cells in TH17_cluster2 as activated (Figure 43). Additionally, the newly detected cells, sharing their clonotype with blood TH17_cluster1 or TH17_cluster2, show stronger marker gene expression of *RORC* and *IL17A*, compared to cells that do not share clonotypes and were not annotated as TH17 cells in the lung. In particular, TH17_cluster2 cells show expression of *RBPJ*, a gene, which is known to control the pathogenicity and activation state of TH17 cells by repressing the production of the anti-inflammatory protein IL-10 and by enhancing the expression of *IL23R*, which is the receptor of the inflammatory cytokine Interleukin-23 (IL-23) [162, 181]. IL-23 is known to enhance TH17 proliferation and is a major regulator of inflammation [181, 182]. Furthermore, there is increasing evidence that TH17 cells play an important role in COVID-19 using various mechanisms of modulating the immune response [181].

However, in comparison with the few patients suffering from pneumonia caused by bacterial infections, there is no strong statistically supported difference in the cell type proportions in COVID-19 of activated TH17 cells (Figure 45). Thus, this data set is too small to allow determining the effect of activated *RBPJ*$^+$ TH17 cells on COVID-19, but it indicates involvement of TH17 cells in severe lung damage. Further *in vitro* and *in vivo* experiments are needed to validate the influence of TH17 cells on lung damage.

4.7 Summary and limitations

Overall, DISCERN showed the best performance across all batch correction and imputation experiments for all three considered data sets. It yields precise and robust reconstructions allowing for unique biological insights in low-quality and high-quality data. The good performance of DISCERN was further validated using application-oriented and literature-based evaluations as well as CITE-seq information. The citeseq data set additionally allowed to employ DISCERN, using a bulk RNA-seq data set as a reference, to detect previously unseen T cell subtypes. Finally, DISCERN was used for detecting two TH17 cell subtypes in the covid-blood data set, which could be verified using clonotype information and a corresponding COVID-19 lung data set. Especially, one of these clusters shows a potential role in the development of severe lung disease triggered by increased TH17 response.

Thus, DISCERN is a valuable tool for the realistic reconstruction of gene expression, which is achieved by the CIN and the two decoder outputs in its neural networks. The CIN enables the evaluation of multiple references with only one trained model of DISCERN, to capture multiple aspects of the data sets. The two decoder outputs are used to disentangle the dropout from the gene expression estimation and together with the sampling procedure, it allows for the realistic generation of scRNA-seq data sets. The ability of DISCERN to use a reference data set enables imputation of missing expression values using measured data.

The source of these missing values can be batch effects and artifacts of the sequencing technology. Since the source of missing values remains largely unknown, DISCERN does not rely on statistical assumptions about the distribution of expression values. While well-based sequencing technologies often provide data with good quality tailored for the question to be solved, bulk RNA-seq usually provides data of high quality in general.

The quality of the bulk RNA-seq data is only limited by the input material, i.e. the cell type purity. However, the availability of public bulk RNA-seq data sets and the generation of such data sets for a corresponding scRNA-seq data set (tissue, species, etc.) can be problematic. While Ota *et al.* [138] made a huge effort to provide a large number of PBMC cell types and samples, no other bulk RNA-seq samples of this size are available for other tissues, but PBMCs. Especially for solid tissues, FACS sorting and subsequent bulk RNA-seq is difficult, due to complex cell dissociation. In these cases, usually snRNA-seq is applied, because only isolation of cell nuclei is required. However, nuclei from solid tissues do not allow for FACS sorting by surface proteins and hence bulk RNA-seq of purified cell types is hardly possible. This lack of data availability of course limits the use of a reconstruction based on bulk RNA-seq data.

4.8 Outlook

DISCERN provides a framework for easy, robust, and precise reconstruction of scRNA-seq data using a reference. The reconstruction with DISCERN is not limited to bulk RNA-seq, but can also be extended to other scRNA-seq data sets to achieve reasonable reconstructions. The Human Cell Atlas [183] is a comprehensive approach to collect reference scRNA-seq data sets for all tissues in the human body. These data sets could potentially be used as a reference data set for the reconstruction with DISCERN for all kinds of tissues. Furthermore, the huge amount of cells sequenced can be used to provide more evidence for low-abundant cell types after integration.

Furthermore, the WAE-architecture of DISCERN allows for operations in the embedding. For example, Lotfollahi, Wolf & Theis [94] used linear operations to predict drug perturbation effects. This was done by computing the relations of untreated cell types and treated cell types in the embedding and applying this perturbation vector to cell types of which the treated state was not measured.

In cases where the disease state was not sequenced, a similar approach could also be used to predict the effect of a disease on cell types. Here findings from Arvanitidis, Hansen & Hauberg [184] would provide a good foundation for non-linear operations in the embedding. This would be compatible with the non-linear relationship between cells often found in a deep-learning-, autoencoder-based embedding [184].

Possible instances of such non-linear functions are geodesics. These could provide a promising way of generating unseen perturbation effects not amendable by linear functions. A geodesic is defined as the shortest path between two points on a surface [185], i.e. the shortest path of two samples with respect to the data. For example, for two points on a sphere, the shortest path using euclidean distance is a straight line through the sphere. In contrast, the geodesic is the shortest path following the surface of the sphere. Geodesics were shown to be a better estimate of the sample distance in autoencoder-based embeddings [184].

On this line, Ding & Regev [186] showed for VAEs that the modification of the prior distribution, i.e. replacing Gaussian with hyperspherical spaces, improved the embedding of scRNA-seq data. Since the WAEs has no constraints for the prior distribution, the DISCERN architecture could easily be extended to use hyperspheres. This could potentially be beneficial for the imputation capabilities by generating improved embeddings which play an important role in the reconstruction of expression data.

These technological methods for generating and analyzing scRNA-seq data were heavily used in thousands of projects in the last years. As the techniques became mature, researchers (e.g. Lance *et al.* [187] and Efremova & Teichmann [188]) began to apply it simultaneously to different types of biological material extracted from the same or similar cells. The resulting multimodal data sets consist, for example, of protein and scRNA-seq in CITE-seq [137], or of chromatin organization and scRNA-seq in ATAC-seq [189] or spatial transcriptomics [190]. However, the reliable measurement of more than two (in rare cases three) modalities is still barely possible [191]. Thus, to obtain information from more modalities, the integration of multiple data sets is required. Recently, autoencoder-based architectures have been used to integrate two data modalities from one [95, 192] or two data sets [193].

Especially the idea of creating modality-specific autoencoders with a shared embedding [193], can easily be incorporated in the current DISCERN framework. The resulting modifications of the DISCERN architecture is likely to considerably improve disentanglement of dropout and expression and thus may allow to more faithfully model scRNA-seq data. The combination of multiple modalities would give a comprehensive overview of single cells and would allow for the reconstruction of absent data points for modalities, which could not be measured simultaneously from a single cell.

Furthermore, the concept of utilizing a high-quality reference to improve low-quality data might be applied to data from other high throughput omics technologies with similar technological limitations as known for scRNA-seq. A premium example of such a technology is single-cell proteomics [194]. In data from single-cell proteomics batch effects occur [195] and values are missing [196] as well.

These technological limitations considerably impede the gain of biological insights. To counter these limitations by using reference-based reconstruction based on deep generative networks, such as the one implemented in DISCERN, is a very promising area of further research. Along these lines, this approach could infer information beyond what is currently measurable. This would truly be transformative in future (biomedical) research.

Bibliography

1. Editorial, N. Method of the year 2013. *Nat. Methods* **11,** 1 (2014).

2. Zhao, Y., Panzer, U., Bonn, S. & Krebs, C. F. Single-cell biology to decode the immune cellular composition of kidney inflammation. *Cell and tissue research* **385.** Publisher: Springer, 435–443 (2021).

3. Wang, X., He, Y., Zhang, Q., Ren, X. & Zhang, Z. Direct comparative analyses of 10X genomics chromium and smart-seq2. *Genomics, proteomics & bioinformatics* **19.** Publisher: Elsevier, 253–266 (2021).

4. Wolfien, M., David, R. & Galow, A.-M. Single-Cell RNA Sequencing Procedures and Data Analysis. *Exon Publications,* 19–35. `https://exonpublications.com/index.php/exon/article/view/265` (Mar. 2021).

5. Ding, J., Adiconis, X., Simmons, S. K., Kowalczyk, M. S., Hession, C. C., Marjanovic, N. D., Hughes, T. K., Wadsworth, M. H., Burks, T., Nguyen, L. T., Kwon, J. Y. H., Barak, B., Ge, W., Kedaigle, A. J., Carroll, S., Li, S., Hacohen, N., Rozenblatt-Rosen, O., Shalek, A. K., Villani, A.-C., Regev, A. & Levin, J. Z. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology* **38,** 737–746. ISSN: 1087-0156, 1546-1696. `https://www.nature.com/articles/s41587-020-0465-8` (June 2020).

6. Mereu, E., Lafzi, A., Moutinho, C., Ziegenhain, C., McCarthy, D. J., Álvarez-Varela, A., Batlle, E., Sagar, Grün, D., Lau, J. K., Boutet, S. C., Sanada, C., Ooi, A., Jones, R. C., Kaihara, K., Brampton, C., Talaga, Y., Sasagawa, Y., Tanaka, K., Hayashi, T., Braeuning, C., Fischer, C., Sauer, S., Trefzer, T., Conrad, C., Adiconis, X., Nguyen, L. T., Regev, A., Levin, J. Z., Parekh, S., Janjic, A., Wange, L. E., Bagnoli, J. W., Enard, W., Gut, M., Sandberg, R., Nikaido, I., Gut, I., Stegle, O. & Heyn, H. Benchmarking single-cell RNA-sequencing protocols for cell atlas projects. *Nature Biotechnology* **38.** Number: 6 Publisher: Nature

Publishing Group, 747–755. ISSN: 1546-1696. https://www.nature.com/articles/s41587-020-0469-4 (June 2020).

7. Sena, J. A., Galotto, G., Devitt, N. P., Connick, M. C., Jacobi, J. L., Umale, P. E., Vidali, L. & Bell, C. J. Unique Molecular Identifiers reveal a novel sequencing artefact with implications for RNA-Seq based gene expression analysis. *Scientific Reports* **8.** Number: 1 Publisher: Nature Publishing Group, 13121. ISSN: 2045-2322. https://www.nature.com/articles/s41598-018-31064-7 (Sept. 2018).
8. Chen, G., Ning, B. & Shi, T. Single-Cell RNA-Seq Technologies and Related Computational Data Analysis. eng. *Frontiers in Genetics* **10,** 317. ISSN: 1664-8021 (2019).
9. Haque, A., Engel, J., Teichmann, S. A. & Lönnberg, T. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. eng. *Genome Medicine* **9,** 75. ISSN: 1756-994X (Aug. 2017).
10. Jovic, D., Liang, X., Zeng, H., Lin, L., Xu, F. & Luo, Y. Single-cell RNA sequencing technologies and applications: A brief overview. *Clinical and Translational Medicine* **12.** ISSN: 2001-1326, 2001-1326. https://onlinelibrary.wiley.com/doi/10.1002/ctm2.694 (Mar. 2022).
11. Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K. & Surani, M. A. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* **6,** 377–382. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/nmeth.1315 (May 2009).
12. Islam, S., Kjällquist, U., Moliner, A., Zajac, P., Fan, J.-B., Lönnerberg, P. & Linnarsson, S. Highly multiplexed and strand-specific single-cell RNA 5′ end sequencing. *Nature Protocols* **7,** 813–828. ISSN: 1754-2189, 1750-2799. https://www.nature.com/articles/nprot.2012.022 (May 2012).
13. Islam, S., Kjällquist, U., Moliner, A., Zajac, P., Fan, J.-B., Lönnerberg, P. & Linnarsson, S. Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Research* **21,** 1160–1167. ISSN: 1088-9051. http://genome.cshlp.org/lookup/doi/10.1101/gr.110882.110 (July 2011).

14. Hashimshony, T., Wagner, F., Sher, N. & Yanai, I. CEL-Seq: Single-Cell RNA-Seq by Multiplexed Linear Amplification. *Cell Reports* **2,** 666–673. ISSN: 22111247. `https://linkinghub.elsevier.com/retrieve/pii/S2211124712002288` (Sept. 2012).

15. Ramsköld, D., Luo, S., Wang, Y.-C., Li, R., Deng, Q., Faridani, O. R., Daniels, G. A., Khrebtukova, I., Loring, J. F., Laurent, L. C., Schroth, G. P. & Sandberg, R. Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nature Biotechnology* **30,** 777–782. ISSN: 1087-0156, 1546-1696. `http://www.nature.com/articles/nbt.2282` (Aug. 2012).

16. Sasagawa, Y., Nikaido, I., Hayashi, T., Danno, H., Uno, K. D., Imai, T. & Ueda, H. R. Quartz-Seq: a highly reproducible and sensitive single-cell RNA sequencing method, reveals non-genetic gene-expression heterogeneity. *Genome Biology* **14,** 3097. ISSN: 1474-760X. `http://genomebiology.biomedcentral.com/articles/10.1186/gb-2013-14-4-r31` (Apr. 2013).

17. Picelli, S., Björklund, Å. K., Faridani, O. R., Sagasser, S., Winberg, G. & Sandberg, R. Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nature Methods* **10,** 1096–1098. ISSN: 1548-7091, 1548-7105. `http://www.nature.com/articles/nmeth.2639` (Nov. 2013).

18. Jaitin, D. A., Kenigsberg, E., Keren-Shaul, H., Elefant, N., Paul, F., Zaretsky, I., Mildner, A., Cohen, N., Jung, S., Tanay, A. & Amit, I. Massively Parallel Single-Cell RNA-Seq for Marker-Free Decomposition of Tissues into Cell Types. *Science* **343,** 776–779. ISSN: 0036-8075, 1095-9203. `https://www.sciencemag.org/lookup/doi/10.1126/science.1247651` (Feb. 2014).

19. Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Sanes, J. R., Shalek, A. K., Regev, A. & McCarroll, S. A. Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161,** 1202–1214. ISSN: 00928674. `https://linkinghub.elsevier.com/retrieve/pii/S0092867415005498` (May 2015).

20. Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A. & Kirschner, M. W. Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell* **161,** 1187–1201. ISSN:

00928674. https://linkinghub.elsevier.com/retrieve/pii/S0092867415005000 (May 2015).

21. Fan, H. C., Fu, G. K. & Fodor, S. P. A. Combinatorial labeling of single cells for gene expression cytometry. *Science* **347,** 1258367. ISSN: 0036-8075, 1095-9203. https://www.science.org/doi/10.1126/science.1258367 (Feb. 2015).

22. Fan, X., Zhang, X., Wu, X., Guo, H., Hu, Y., Tang, F. & Huang, Y. Single-cell RNA-seq transcriptome analysis of linear and circular RNAs in mouse preimplantation embryos. *Genome Biology* **16,** 148. ISSN: 1465-6906. https://genomebiology.biomedcentral.com/articles/10.1186/s13059-015-0706-1 (Dec. 2015).

23. Hashimshony, T., Senderovich, N., Avital, G., Klochendler, A., de Leeuw, Y., Anavy, L., Gennert, D., Li, S., Livak, K. J., Rozenblatt-Rosen, O., Dor, Y., Regev, A. & Yanai, I. CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq. *Genome Biology* **17,** 77. ISSN: 1474-760X. http://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0938-8 (Dec. 2016).

24. Xin, Y., Kim, J., Ni, M., Wei, Y., Okamoto, H., Lee, J., Adler, C., Cavino, K., Murphy, A. J., Yancopoulos, G. D., Lin, H. C. & Gromada, J. Use of the Fluidigm C1 platform for RNA sequencing of single mouse pancreatic islet cells. eng. *Proceedings of the National Academy of Sciences of the United States of America* **113,** 3293–3298. ISSN: 1091-6490 (Mar. 2016).

25. Zheng, G. X. Y., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., Gregory, M. T., Shuga, J., Montesclaros, L., Underwood, J. G., Masquelier, D. A., Nishimura, S. Y., Schnall-Levin, M., Wyatt, P. W., Hindson, C. M., Bharadwaj, R., Wong, A., Ness, K. D., Beppu, L. W., Deeg, H. J., McFarland, C., Loeb, K. R., Valente, W. J., Ericson, N. G., Stevens, E. A., Radich, J. P., Mikkelsen, T. S., Hindson, B. J. & Bielas, J. H. Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **8,** 14049. ISSN: 2041-1723. http://www.nature.com/articles/ncomms14049 (Apr. 2017).

26. Habib, N., Avraham-Davidi, I., Basu, A., Burks, T., Shekhar, K., Hofree, M., Choudhury, S. R., Aguet, F., Gelfand, E., Ardlie, K., Weitz, D. A., Rozenblatt-Rosen, O., Zhang, F. & Regev, A. Massively parallel single-nucleus RNA-

seq with DroNc-seq. *Nature Methods* **14,** 955–958. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/nmeth.4407 (Oct. 2017).

27. Cao, J., Packer, J. S., Ramani, V., Cusanovich, D. A., Huynh, C., Daza, R., Qiu, X., Lee, C., Furlan, S. N., Steemers, F. J., Adey, A., Waterston, R. H., Trapnell, C. & Shendure, J. Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* **357,** 661–667. ISSN: 0036-8075, 1095-9203. https://www.science.org/doi/10.1126/science.aam8940 (Aug. 2017).

28. Gierahn, T. M., Wadsworth, M. H., Hughes, T. K., Bryson, B. D., Butler, A., Satija, R., Fortune, S., Love, J. C. & Shalek, A. K. Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nature Methods* **14,** 395–398. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/nmeth.4179 (Apr. 2017).

29. Sheng, K., Cao, W., Niu, Y., Deng, Q. & Zong, C. Effective detection of variation in single-cell transcriptomes using MATQ-seq. *Nature Methods* **14,** 267–270. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/nmeth.4145 (Mar. 2017).

30. Rosenberg, A. B., Roco, C. M., Muscat, R. A., Kuchina, A., Sample, P., Yao, Z., Graybuck, L. T., Peeler, D. J., Mukherjee, S., Chen, W., Pun, S. H., Sellers, D. L., Tasic, B. & Seelig, G. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360,** 176–182. ISSN: 0036-8075, 1095-9203. https://www.science.org/doi/10.1126/science.aam8999 (Apr. 2018).

31. Sasagawa, Y., Danno, H., Takada, H., Ebisawa, M., Tanaka, K., Hayashi, T., Kurisaki, A. & Nikaido, I. Quartz-Seq2: a high-throughput single-cell RNA-sequencing method that effectively uses limited sequence reads. *Genome Biology* **19,** 29. ISSN: 1474-760X. https://genomebiology.biomedcentral.com/articles/10.1186/s13059-018-1407-3 (Dec. 2018).

32. Liu, C., Wu, T., Fan, F., Liu, Y., Wu, L., Junkin, M., Wang, Z., Yu, Y., Wang, W., Wei, W., Yuan, Y., Wang, M., Cheng, M., Wei, X., Xu, J., Shi, Q., Liu, S., Chen, A., Wang, O., Ni, M., Zhang, W., Shang, Z., Lai, Y., Guo, P., Ward, C., Volpe, G., Wang, L., Zheng, H., Liu, Y., Peters, B. A., Beecher, J., Zhang, Y., Esteban, M. A., Hou, Y., Xu, X., Chen, I.-J. & Liu, L. *A portable and cost-effective microfluidic system for massively parallel single-cell transcriptome profiling* Pages:

818450 Section: New Results. Nov. 2019. https://www.biorxiv.org/content/10.1101/818450v3.

33. Oller-Moreno, S., Kloiber, K., Machart, P. & Bonn, S. Algorithmic advances in machine learning for single-cell expression analysis. *Current Opinion in Systems Biology* **25.** Publisher: Elsevier, 27–33 (2021).

34. Lähnemann, D., Köster, J., Szczurek, E., McCarthy, D. J., Hicks, S. C., Robinson, M. D., Vallejos, C. A., Campbell, K. R., Beerenwinkel, N., Mahfouz, A., Pinello, L., Skums, P., Stamatakis, A., Attolini, C. S.-O., Aparicio, S., Baaijens, J., Balvert, M., Barbanson, B. d., Cappuccio, A., Corleone, G., Dutilh, B. E., Florescu, M., Guryev, V., Holmer, R., Jahn, K., Lobo, T. J., Keizer, E. M., Khatri, I., Kielbasa, S. M., Korbel, J. O., Kozlov, A. M., Kuo, T.-H., Lelieveldt, B. P., Mandoiu, I. I., Marioni, J. C., Marschall, T., Mölder, F., Niknejad, A., Raczkowski, L., Reinders, M., Ridder, J. d., Saliba, A.-E., Somarakis, A., Stegle, O., Theis, F. J., Yang, H., Zelikovsky, A., McHardy, A. C., Raphael, B. J., Shah, S. P. & Schönhuth, A. Eleven grand challenges in single-cell data science. *Genome Biology* **21,** 31. ISSN: 1474-760X. https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-1926-6 (Dec. 2020).

35. Shalek, A. K., Satija, R., Shuga, J., Trombetta, J. J., Gennert, D., Lu, D., Chen, P., Gertner, R. S., Gaublomme, J. T., Yosef, N., Schwartz, S., Fowler, B., Weaver, S., Wang, J., Wang, X., Ding, R., Raychowdhury, R., Friedman, N., Hacohen, N., Park, H., May, A. P. & Regev, A. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* **510,** 363–369. ISSN: 0028-0836, 1476-4687. http://www.nature.com/articles/nature13437 (June 2014).

36. Kim, T. H., Zhou, X. & Chen, M. Demystifying "drop-outs" in single-cell UMI data. *Genome Biology* **21,** 196. ISSN: 1474-760X. https://doi.org/10.1186/s13059-020-02096-y (Aug. 2020).

37. Jiang, R., Sun, T., Song, D. & Li, J. J. Statistics or biology: the zero-inflation controversy about scRNA-seq data. *Genome Biology* **23,** 31. ISSN: 1474-760X. https://doi.org/10.1186/s13059-022-02601-5 (Jan. 2022).

38. Hou, W., Ji, Z., Ji, H. & Hicks, S. C. A systematic evaluation of single-cell RNA-sequencing imputation methods. *Genome Biology* **21,** 218. ISSN: 1474-

760X. https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-02132-x (Dec. 2020).

39. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications* **10.** Publisher: Nature Publishing Group, 1–14 (2019).

40. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications* **9.** Publisher: Nature Publishing Group, 1–9 (2018).

41. Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., Bierie, B., Mazutis, L., Wolf, G., Krishnaswamy, S. & Pe'er, D. Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell* **174,** 716–729.e27. ISSN: 00928674. https://linkinghub.elsevier.com/retrieve/pii/S0092867418307244 (July 2018).

42. Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X. & Garmire, L. X. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome biology* **20.** Publisher: BioMed Central, 1–14 (2019).

43. Lakkis, J., Wang, D., Zhang, Y., Hu, G., Wang, K., Pan, H., Ungar, L., Reilly, M. P., Li, X. & Li, M. A joint deep learning model for simultaneous batch effect correction, denoising and clustering in single-cell transcriptomics. *bioRxiv.* Publisher: Cold Spring Harbor Laboratory (2020).

44. Huizing, G.-J., Peyré, G. & Cantini, L. Optimal transport improves cell–cell similarity inference in single-cell omics data. *Bioinformatics* **38,** 2169–2177. ISSN: 1367-4803. https://doi.org/10.1093/bioinformatics/btac084 (Apr. 2022).

45. Slyper, M., Porter, C. B. M., Ashenberg, O., Waldman, J., Drokhlyansky, E., Wakiro, I., Smillie, C., Smith-Rosario, G., Wu, J., Dionne, D., Vigneau, S., Jané-Valbuena, J., Tickle, T. L., Napolitano, S., Su, M.-J., Patel, A. G., Karlstrom, A., Gritsch, S., Nomura, M., Waghray, A., Gohil, S. H., Tsankov, A. M., Jerby-Arnon, L., Cohen, O., Klughammer, J., Rosen, Y., Gould, J., Nguyen, L., Hofree, M., Tramontozzi, P. J., Li, B., Wu, C. J., Izar, B., Haq, R., Hodi, F. S., Yoon, C. H., Hata, A. N., Baker, S. J., Suvà, M. L., Bueno, R., Stover,

E. H., Clay, M. R., Dyer, M. A., Collins, N. B., Matulonis, U. A., Wagle, N., Johnson, B. E., Rotem, A., Rozenblatt-Rosen, O. & Regev, A. A single-cell and single-nucleus RNA-Seq toolbox for fresh and frozen human tumors. *Nature Medicine* **26,** 792–802. ISSN: 1078-8956, 1546-170X. http://www.nature.com/articles/s41591-020-0844-1 (May 2020).

46. Wen, Z.-H., Langsam, J. L., Zhang, L., Shen, W. & Zhou, X. A Bayesian factorization method to recover single-cell RNA sequencing data. *Cell reports methods* **2.** Publisher: Elsevier, 100133 (2022).

47. Peng, T., Zhu, Q., Yin, P. & Tan, K. SCRABBLE: single-cell RNA-seq imputation constrained by bulk RNA-seq data. *Genome biology* **20.** Publisher: Springer, 1–12 (2019).

48. Hu, Z., Zu, S. & Liu, J. S. SIMPLEs: a single-cell RNA sequencing imputation strategy preserving gene modules and cell clusters variation. *NAR genomics and bioinformatics* **2.** Publisher: Oxford University Press, lqaa077 (2020).

49. Tran, H. T. N., Ang, K. S., Chevrier, M., Zhang, X., Lee, N. Y. S., Goh, M. & Chen, J. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. eng. *Genome Biology* **21,** 12. ISSN: 1474-760X (Jan. 2020).

50. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology* **36.** Number: 5 Publisher: Nature Publishing Group, 421–427. ISSN: 1546-1696. https://www.nature.com/articles/nbt.4091 (May 2018).

51. Sprang, M., Andrade-Navarro, M. A. & Fontaine, J.-F. Batch effect detection and correction in RNA-seq data using machine-learning-based automated assessment of quality. *BMC Bioinformatics* **23,** 279. ISSN: 1471-2105. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-022-04775-y (July 2022).

52. Goh, W. W. B., Wang, W. & Wong, L. Why Batch Effects Matter in Omics Data, and How to Avoid Them. *Trends in Biotechnology* **35,** 498–507. ISSN: 01677799. https://linkinghub.elsevier.com/retrieve/pii/S0167779917300367 (June 2017).

53. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., Hao, Y., Stoeckius, M., Smibert, P. & Satija, R. Comprehensive integration of single-cell data. *Cell* **177.** Publisher: Elsevier, 1888–1902 (2019).

54. Lotfollahi, M., Naghipourfar, M., Luecken, M. D., Khajavi, M., Büttner, M., Wagenstetter, M., Avsec, Ž., Gayoso, A., Yosef, N., Interlandi, M., Rybakov, S., Misharin, A. V. & Theis, F. J. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology* **40,** 121–130. ISSN: 1087-0156, 1546-1696. `https://www.nature.com/articles/s41587-021-01001-7` (Jan. 2022).

55. Gayoso, A., Lopez, R., Xing, G., Boyeau, P., Valiollah Pour Amiri, V., Hong, J., Wu, K., Jayasuriya, M., Mehlman, E., Langevin, M., Liu, Y., Samaran, J., Misrachi, G., Nazaret, A., Clivio, O., Xu, C., Ashuach, T., Gabitto, M., Lotfollahi, M., Svensson, V., da Veiga Beltrame, E., Kleshchevnikov, V., Talavera-López, C., Pachter, L., Theis, F. J., Streets, A., Jordan, M. I., Regier, J. & Yosef, N. A Python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology* **40,** 163–166. ISSN: 1087-0156, 1546-1696. `https://www.nature.com/articles/s41587-021-01206-w` (Feb. 2022).

56. Peng, L., Chen, Y., Ou, Q., Wang, X. & Tang, N. LncRNA MIAT correlates with immune infiltrates and drug reactions in hepatocellular carcinoma. *International immunopharmacology* **89.** Publisher: Elsevier, 107071 (2020).

57. Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-r. & Raychaudhuri, S. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods* **16.** Number: 12 Publisher: Nature Publishing Group, 1289–1296. ISSN: 1548-7105. `https://www.nature.com/articles/s41592-019-0619-0` (Dec. 2019).

58. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature Methods* **15,** 1053–1058. ISSN: 1548-7091, 1548-7105. `http://www.nature.com/articles/s41592-018-0229-2` (Dec. 2018).

59. Cui, X. & Churchill, G. A. Statistical tests for differential expression in cDNA microarray experiments. *Genome Biology* **4,** 210. ISSN: 1474-760X. `https://genomebiology.biomedcentral.com/articles/10.1186/gb-2003-4-4-210` (Apr. 2003).

60. Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S. & Mesirov, J. P. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* **102,** 15545–15550. ISSN: 0027-8424, 1091-6490. `https://pnas.org/doi/full/10.1073/pnas.0506580102` (Oct. 2005).

61. Chen, E. Y., Tan, C. M., Kou, Y., Duan, Q., Wang, Z., Meirelles, G. V., Clark, N. R. & Ma'ayan, A. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC bioinformatics* **14.** Publisher: BioMed Central, 1–14 (2013).

62. Gene Ontology Consortium. The Gene Ontology resource: enriching a GOld mine. eng. *Nucleic Acids Research* **49,** D325–D334. ISSN: 1362-4962 (Jan. 2021).

63. Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H. & Kanehisa, M. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* **27,** 29–34. ISSN: 0305-1048, 1362-4962. `https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/27.1.29` (Jan. 1999).

64. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9.** ISSN: 1532-4435 (2008).

65. McInnes, L., Healy, J. & Melville, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction* arXiv:1802.03426 [cs, stat]. Sept. 2020. `http://arxiv.org/abs/1802.03426`.

66. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology* **15.** ISSN: 1744-4292, 1744-4292. `https://onlinelibrary.wiley.com/doi/10.15252/msb.20188746` (June 2019).

67. Csiszar, I. I-Divergence Geometry of Probability Distributions and Minimization Problems. *The Annals of Probability* **3.** ISSN: 0091-1798. `https://projecteuclid.org/journals/annals-of-probability/volume-3/issue-1/I-Divergence-Geometry-of-Probability-Distributions-and-Minimization-Problems/10.1214/aop/1176996454.full` (Feb. 1975).

68. Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W. H., Ng, L. G., Ginhoux, F. & Newell, E. W. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology* **37,** 38–44. ISSN: 1087-0156, 1546-1696. `http://www.nature.com/articles/nbt.4314` (Jan. 2019).

69. Kobak, D. & Linderman, G. C. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology* **39,** 156–157. ISSN: 1087-0156, 1546-1696. `http://www.nature.com/articles/s41587-020-00809-z` (Feb. 2021).

70. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **20.** Publisher: Elsevier, 53–65 (1987).

71. Hubert, L. & Arabie, P. Comparing partitions. *Journal of classification* **2.** Publisher: Springer, 193–218 (1985).

72. Vinh, N. X., Epps, J. & Bailey, J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *Journal of Machine Learning Research,* 2837–2854 (2010).

73. Jebara, T. *Machine Learning* ISBN: 978-1-4613-4756-9. `http://link.springer.com/10.1007/978-1-4419-9011-2` (Springer US, Boston, MA, 2004).

74. Lopez, R., Gayoso, A. & Yosef, N. Enhancing scientific discoveries in molecular biology with deep generative models. *Molecular Systems Biology* **16.** Publisher: John Wiley & Sons, Ltd, e9198. ISSN: 1744-4292. `https://www.embopress.org/doi/full/10.15252/msb.20199198` (Sept. 2020).

75. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. Generative Adversarial Nets, 9 (2014).

76. Kurutach, T., Tamar, A., Yang, G., Russell, S. J. & Abbeel, P. Learning plannable representations with causal infogan. *Advances in Neural Information Processing Systems* **31** (2018).

77. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J. & Wang, Z. *Photo-realistic single image super-resolution using a generative adversarial network* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 4681–4690.

78. Sanchez-Lengeling, B., Outeiral, C., Guimaraes, G. L. & Aspuru-Guzik, A. *Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC)* preprint (Chemistry, Aug. 2017). https://chemrxiv.org/engage/chemrxiv/article-details/60c73d91702a9beea7189bc2.

79. Marouf, M., Machart, P., Bansal, V., Kilian, C., Magruder, D. S., Krebs, C. F. & Bonn, S. Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. *Nature communications* **11.** Publisher: Nature Publishing Group, 1–12 (2020).

80. Yu, H. & Welch, J. D. MichiGAN: sampling from disentangled representations of single-cell data using generative adversarial networks. *Genome biology* **22,** 1–26 (2021).

81. Xu, Y., Zhang, Z., You, L., Liu, J., Fan, Z. & Zhou, X. scIGANs: single-cell RNA-seq imputation using generative adversarial networks. *Nucleic Acids Research* **48,** e85. ISSN: 0305-1048. https://doi.org/10.1093/nar/gkaa506 (Sept. 2020).

82. Hinton, G. E. & Zemel, R. Autoencoders, minimum description length and Helmholtz free energy. *Advances in neural information processing systems* **6** (1993).

83. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. eng. *Science (New York, N.Y.)* **313,** 504–507. ISSN: 1095-9203 (July 2006).

84. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* Google-Books-ID: omivDQAAQBAJ. ISBN: 978-0-262-33737-3 (MIT Press, Nov. 2016).

85. Lopez Pinaya, W. H., Vieira, S., Garcia-Dias, R. & Mechelli, A. in *Machine Learning* 193–208 (Elsevier, 2020). ISBN: 978-0-12-815739-8. https://linkinghub.elsevier.com/retrieve/pii/B9780128157398000110.

86. Fournier, Q. & Aloise, D. *Empirical comparison between autoencoders and traditional dimensionality reduction methods* in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)* arXiv:2103.04874 [cs] (June 2019), 211–214. http://arxiv.org/abs/2103.04874.

87. Ohno, H. Auto-encoder-based generative models for data augmentation on regression problems. *Soft Computing* **24,** 7999–8009. ISSN: 1432-7643, 1433-7479. http://link.springer.com/10.1007/s00500-019-04094-0 (June 2020).

88. Kingma, D. P. & Welling, M. *Auto-Encoding Variational Bayes* arXiv:1312.6114 [cs, stat]. May 2014. http://arxiv.org/abs/1312.6114.

89. Pinheiro Cinelli, L., Araújo Marins, M., Barros da Silva, E. A. & Lima Netto, S. in *Variational Methods for Machine Learning with Applications to Deep Networks* 111–149 (Springer International Publishing, Cham, 2021). ISBN: 978-3-030-70678-4. https://link.springer.com/10.1007/978-3-030-70679-1_5.

90. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323,** 533–536. ISSN: 0028-0836, 1476-4687. http://www.nature.com/articles/323533a0 (Oct. 1986).

91. Tolstikhin, I., Bousquet, O., Gelly, S. & Schoelkopf, B. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558* (2017).

92. Theis, L., Oord, A. v. d. & Bethge, M. *A note on the evaluation of generative models* arXiv:1511.01844 [cs, stat]. Apr. 2016. http://arxiv.org/abs/1511.01844.

93. Xu, C., Lopez, R., Mehlman, E., Regier, J., Jordan, M. I. & Yosef, N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular Systems Biology* **17.** ISSN: 1744-4292, 1744-4292. https://onlinelibrary.wiley.com/doi/10.15252/msb.20209620 (Jan. 2021).

94. Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nature methods* **16.** Publisher: Nature Publishing Group, 715–721 (2019).

95. Lotfollahi, M., Litinetskaya, A. & Theis, F. J. Multigrate: single-cell multi-omic data integration. *bioRxiv.* Publisher: Cold Spring Harbor Laboratory (2022).

96. Schriever, H. & Kostka, D. Vaeda computationally annotates doublets in single-cell RNA sequencing data. *Bioinformatics* (ed Mathelier, A.) btac720. ISSN: 1367-4803, 1367-4811. https://academic.oup.com/bioinformatics/

advance-article/doi/10.1093/bioinformatics/btac720/6808614 (Nov. 2022).

97. Villani, C. *Optimal transport: old and new* (Springer, 2009).

98. Arjovsky, M., Chintala, S. & Bottou, L. *Wasserstein generative adversarial networks* in *International conference on machine learning* (PMLR, 2017), 214–223.

99. Kantorovich, L. V. Mathematical Methods of Organizing and Planning Production. *Management Science* **6,** 366–422. ISSN: 0025-1909, 1526-5501. http://pubsonline.informs.org/doi/10.1287/mnsc.6.4.366 (July 1960).

100. Vaserstein, L. N. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii* **5.** Publisher: Russian Academy of Sciences, Branch of Informatics, Computer Equipment and . . ., 64–72 (1969).

101. Monge, G. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.,* 666–704 (1781).

102. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. A Kernel Two-Sample Test. *Journal of Machine Learning Research* **13,** 723–773. http://jmlr.org/papers/v13/gretton12a.html (2012).

103. Berlinet, A. & Thomas-Agnan, C. *Reproducing Kernel Hilbert Spaces in Probability and Statistics* ISBN: 978-1-4613-4792-7. http://link.springer.com/10.1007/978-1-4419-9096-9 (Springer US, Boston, MA, 2004).

104. Rubenstein, P. K., Schoelkopf, B. & Tolstikhin, I. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761* (2018).

105. Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Wolf, F. A. Conditional out-of-distribution generation for unpaired data using transfer VAE. *Bioinformatics* **36.** Publisher: Oxford University Press, i610–i617 (2020).

106. Luecken, M. D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Mueller, M. F., Strobl, D. C., Zappia, L., Dugas, M., Colomé-Tatché, M. & Theis, F. J. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods* **19,** 41–50. ISSN: 1548-7091, 1548-7105. https://www.nature.com/articles/s41592-021-01336-8 (Jan. 2022).

107. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* arXiv:1502.03167 [cs]. Mar. 2015. `http://arxiv.org/abs/1502.03167`.

108. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

109. Dumoulin, V., Shlens, J. & Kudlur, M. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629* (2016).

110. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Las Vegas, NV, USA, June 2016), 770–778. ISBN: 978-1-4673-8851-1. `http://ieeexplore.ieee.org/document/7780459/`.

111. Simon, M., Rodner, E. & Denzler, J. *ImageNet pre-trained models with batch normalization* arXiv:1612.01452 [cs]. Dec. 2016. `http://arxiv.org/abs/1612.01452`.

112. Santurkar, S., Tsipras, D., Ilyas, A. & Madry, A. *How Does Batch Normalization Help Optimization?* arXiv:1805.11604 [cs, stat]. Apr. 2019. `http://arxiv.org/abs/1805.11604`.

113. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. *Attention Is All You Need* arXiv:1706.03762 [cs]. Dec. 2017. `http://arxiv.org/abs/1706.03762`.

114. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P. & Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature* **596,** 583–589. ISSN: 0028-0836, 1476-4687. `https://www.nature.com/articles/s41586-021-03819-2` (Aug. 2021).

115. Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. *F1000Research* **7.** Publisher: Faculty of 1000 Ltd (2018).

116. Hancock, J. T. & Khoshgoftaar, T. M. Survey on categorical data for neural networks. *Journal of Big Data* **7,** 28. ISSN: 2196-1115. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00305-w (Dec. 2020).

117. Vanhoucke, V., Senior, A. & Mao, M. Z. *Improving the speed of neural networks on CPUs* in *Deep learning and unsupervised feature learning workshop, NIPS 2011* (2011).

118. Lavda, F., Gregorová, M. & Kalousis, A. *Improving VAE generations of multimodal data through data-dependent conditional priors* arXiv:1911.10885 [cs, stat]. Nov. 2019. http://arxiv.org/abs/1911.10885.

119. Huber, P. J. Robust estimation of a location parameter. *Annals Mathematics Statistics* (1964).

120. Gayoso, A., Lopez, R., Xing, G., Boyeau, P., Wu, K., Jayasuriya, M., Melhman, E., Langevin, M., Liu, Y., Samaran, J., *et al.* Scvi-tools: A library for deep probabilistic analysis of single-cell omics data. *bioRxiv.* Publisher: Cold Spring Harbor Laboratory (2021).

121. Misra, D. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681* (2019).

122. Le, L., Patterson, A. & White, M. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. *Advances in neural information processing systems* **31** (2018).

123. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15,** 1929–1958. ISSN: 1533-7928. http://jmlr.org/papers/v15/srivastava14a.html (2014).

124. Zhang, J., Ji, N., Liu, J., Pan, J. & Meng, D. Enhancing performance of the backpropagation algorithm via sparse response regularization. *Neurocomputing* **153,** 20–40. ISSN: 09252312. https://linkinghub.elsevier.com/retrieve/pii/S092523121401649X (Apr. 2015).

125. Shi, G., Zhang, J., Li, H. & Wang, C. Enhance the Performance of Deep Neural Networks via L2 Regularization on the Input of Activations. *Neural Processing Letters* **50,** 57–75. ISSN: 1370-4621, 1573-773X. http://link.springer.com/10.1007/s11063-018-9883-8 (Aug. 2019).

126. Girosi, F., Jones, M. & Poggio, T. Regularization theory and neural networks architectures. *Neural computation* **7.** Publisher: MIT Press, 219–269. ISSN: 0899-7667 (1995).

127. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J. & Han, J. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265* (2019).

128. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

129. Popel, M. & Bojar, O. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics* **110.** arXiv:1804.00247 [cs], 43–70. ISSN: 1804-0462. `http://arxiv.org/abs/1804.00247` (Apr. 2018).

130. Feurer, M. & Hutter, F. in *Automated Machine Learning* (eds Hutter, F., Kotthoff, L. & Vanschoren, J.) Series Title: The Springer Series on Challenges in Machine Learning, 3–33 (Springer International Publishing, Cham, 2019). ISBN: 978-3-030-05317-8. `http://link.springer.com/10.1007/978-3-030-05318-5_1`.

131. Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S. & Smith, K. Cython: The Best of Both Worlds. *Computing in Science & Engineering* **13,** 31–39. ISSN: 1521-9615. `http://ieeexplore.ieee.org/document/5582062/` (Mar. 2011).

132. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E. & Stoica, I. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118* (2018).

133. Justus, D., Brennan, J., Bonner, S. & McGough, A. S. *Predicting the Computational Cost of Deep Learning Models* in *2018 IEEE International Conference on Big Data (Big Data)* (IEEE, Seattle, WA, USA, Dec. 2018), 3873–3882. ISBN: 978-1-5386-5035-6. `https://ieeexplore.ieee.org/document/8622396/`.

134. Bisong, E. in *Building Machine Learning and Deep Learning Models on Google Cloud Platform* 203–207 (Apress, Berkeley, CA, 2019). ISBN: 978-1-4842-4469-2. `http://link.springer.com/10.1007/978-1-4842-4470-8_16`.

135. Lab, S. *panc8.SeuratData: Eight Pancreas Datasets Across Five Technologies* (2019).

136. Linderman, G. C., Zhao, J., Roulis, M., Bielecki, P., Flavell, R. A., Nadler, B. & Kluger, Y. Zero-preserving imputation of single-cell RNA-seq data. *Nature Communications* **13.** Publisher: Nature Publishing Group, 1–11 (2022).

137. Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., Satija, R. & Smibert, P. Simultaneous epitope and transcriptome measurement in single cells. *Nature methods* **14.** Publisher: Nature Publishing Group, 865–868 (2017).

138. Ota, M., Nagafuchi, Y., Hatano, H., Ishigaki, K., Terao, C., Takeshima, Y., Yanaoka, H., Kobayashi, S., Okubo, M., Shirai, H., Sugimori, Y., Maeda, J., Nakano, M., Yamada, S., Yoshida, R., Tsuchiya, H., Tsuchida, Y., Akizuki, S., Yoshifuji, H., Ohmura, K., Mimori, T., Yoshida, K., Kurosaka, D., Okada, M., Setoguchi, K., Kaneko, H., Ban, N., Yabuki, N., Matsuki, K., Mutoh, H., Oyama, S., Okazaki, M., Tsunoda, H., Iwasaki, Y., Sumitomo, S., Shoda, H., Kochi, Y., Okada, Y., Yamamoto, K., Okamura, T. & Fujio, K. Dynamic landscape of immune cell-specific gene regulation in immune-mediated diseases. *Cell* **184,** 3006–3021.e17. ISSN: 00928674. `https://linkinghub.elsevier.com/retrieve/pii/S0092867421004293` (May 2021).

139. Menon, R., Bomback, A. S., Lake, B. B., Stutzke, C., Grewenow, S. M., Menez, S., D'Agati, V. D., Jain, S., Knight, R., Lecker, S. H., Stillman, I., Bogen, S., Beck, L. H., Waikar, S., McMahon, G. M., Weins, A., Colona, M. R., Hacohen, N., Hoover, P. J., Aulisio, M., Bush, W. S., Crawford, D. C., O'toole, J., Poggio, E., Sedor, J., Cooperman, L., Jolly, S., Herlitz, L., Nguyen, J., Gonzalez-Vicente, A., Palmer, E., Sendrey, D., Vinovskis, C., Bjornstad, P. M., Appelbaum, P., Barasch, J. M., Bomback, A. S., D'Agati, V. D., Kiryluk, K., Mehl, K., Canetta, P. A., Shang, N., Balderes, O., Kudose, S., Bansal, S., Alexandrov, T., Rennke, H., El-Achkar, T. M., Cheng, Y., Dagher, P. C., Eadon, M. T., Dunn, K. W., Kelly, K. J., Sutton, T. A., Barwinska, D., Ferkowicz, M. J., Winfree, S., Bledsoe, S., Rivera, M., Williams, J. C., Ferreira, R. M., Parikh, C. R., Corona-Villalobos, C. P., Menez, S., Rosenberg, A., Rosas, S. E., Roy, N., Williams, M., Azeloglu, E. U., He, C., Iyengar, R., Hansen, J., Xiong, Y., Rovin, B., Parikh, S., Shapiro, J. P., Anderton, C. R., Pasa-Tolic, L., Velickovic, D., Lukowski, J., Oliver, G., Ardayfio, J., Bebiak, J., Brown, K., Campbell, C. E., Saul, J., Shpigel, A., Stutzke, C., Koewler, R., Campbell, T., Hayashi, L., Jefferson, N., Roberts, G. V., Pinkeney, R., Troyanskaya, O., Sealfon, R., Tuttle, K. R., Goltsev, Y., Zhang, K., Lake, B. B., Laszik, Z. G., Nolan, G., Boada, P., Sarwal, M., Sigdel, T., Lee, P. J., Alloway, R. R., Woodle, E. S., Ascani, H., Balis, U. G., Hodgin, J. B., Kretzler, M., Lienczewski, C., Mariani, L. H., Menon, R., Steck, B., He, Y., Otto, E., Schaub, J., Blanc, V. M., Eddy, S., Conser, N. C., Luo, J.,

Palevsky, P. M., Rosengart, M., Kellum, J. A., Hall, D. E., Randhawa, P., Tublin, M., Murugan, R., Elder, M. M., Winters, J., Alpers, C. E., Blank, K. N., Carson, J., De Boer, I. H., Dighe, A. L., Himmelfarb, J., Mooney, S. D., Shankland, S., Williams, K., Park, C., Dowd, F., McClelland, R. L., Daniel, S., Hoofnagle, A. N., Wilcox, A., Grewenow, S. M., Bansal, S., Sharma, K., Venkatachalam, M., Zhang, G., Pamreddy, A., Ye, H., Montellano, R., Toto, R. D., Vazquez, M., Lee, S. C., Miller, R. T., Moe, O. W., Torrealba, J., Wang, N., Kermani, A., Sambandam, K., Park, H., Hedayati, S. S., Lu, C. Y., Jain, S., Vijayan, A., Gaut, J. P., Moledina, D., Wilson, F. P., Ugwuowo, U. & Arora, T. Integrated single-cell sequencing and histopathological analyses reveal diverse injury and repair responses in a participant with acute kidney injury: a clinical-molecular-pathologic correlation. *Kidney International* **101,** 1116–1125. ISSN: 00852538. `https://linkinghub.elsevier.com/retrieve/pii/S0085253822002150` (June 2022).

140. Zhao, Y., Kilian, C., Turner, J.-E., Bosurgi, L., Roedl, K., Bartsch, P., Gnirck, A.-C., Cortesi, F., Schultheiß, C., Hellmig, M., Enk, L. U., Hausmann, F., Borchers, A., Wong, M. N., Paust, H.-J., Siracusa, F., Scheibel, N., Herrmann, M., Rosati, E., Bacher, P., Kylies, D., Jarczak, D., Lütgehetmann, M., Pfefferle, S., Steurer, S., Schulze zur Wiesch, J., Puelles, V. G., Sperhake, J.-P., Addo, M. M., Lohse, A. W., Binder, M., Huber, S., Huber, T. B., Kluge, S., Bonn, S., Panzer, U., Gagliani, N. & Krebs, C. F. Clonal expansion and activation of tissue-resident memory-like T $_H$ 17 cells expressing GM-CSF in the lungs of patients with severe COVID-19. *Science Immunology* **6,** eabf6692. ISSN: 2470-9468. `https://www.science.org/doi/10.1126/sciimmunol.abf6692` (Feb. 2021).

141. Traag, V. A., Waltman, L. & Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* **9.** Publisher: Nature Publishing Group, 1–12 (2019).

142. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* **19.** Publisher: Springer, 1–5 (2018).

143. Romano, S., Vinh, N. X., Bailey, J. & Verspoor, K. Adjusting for chance clustering comparison measures. *The Journal of Machine Learning Research* **17.** Publisher: JMLR. org, 4635–4666. ISSN: 1532-4435 (2016).

144. KEGG PATHWAY Database. *T cell receptor signaling pathway - Homo sapiens (human)* 2022. `https://www.genome.jp/pathway/hsa04660`.

145. KEGG PATHWAY Database. *Antigen processing and presentation - Homo sapiens (human)* 2022. `https://www.genome.jp/pathway/hsa04660`.

146. Ortmann, B., Copeman, J., Lehner, P. J., Sadasivan, B., Herberg, J. A., Grandea, A. G., Riddell, S. R., Tampé, R., Spies, T., Trowsdale, J. & Cresswell, P. A critical role for tapasin in the assembly and function of multimeric MHC class I-TAP complexes. eng. *Science (New York, N.Y.)* **277,** 1306–1309. ISSN: 0036-8075 (Aug. 1997).

147. Fu, Z., R Gilbert, E. & Liu, D. Regulation of insulin synthesis and secretion and pancreatic Beta-cell dysfunction in diabetes. *Current diabetes reviews* **9.** Publisher: Bentham Science Publishers, 25–53 (2013).

148. Said, S., Kurtin, P. J., Nasr, S. H., Graham, R. P., Dasari, S., Vrana, J. A., Yasir, S., Torbenson, M. S., Zhang, L., Mounajjed, T., Eric Chen, Z.-M., Lee, H. E. & Wu, T.-T. Carboxypeptidase A1 and regenerating islet-derived 1alpha as new markers for pancreatic acinar cell carcinoma. *Human Pathology* **103,** 120–126. ISSN: 00468177. `https://linkinghub.elsevier.com/retrieve/pii/S004681772030143X` (Sept. 2020).

149. Braun, M. The somatostatin receptor in human pancreatic beta-cells. *Vitamins & Hormones* **95.** Publisher: Elsevier, 165–193 (2014).

150. Rajasree Menon. *Personal communication with Rajasree Menon* 2022.

151. Hansen, J., Sealfon, R., Menon, R., Eadon, M. T., Lake, B. B., Steck, B., Anjani, K., Parikh, S., Sigdel, T. K., Zhang, G., Velickovic, D., Barwinska, D., Alexandrov, T., Dobi, D., Rashmi, P., Otto, E. A., Rivera, M., Rose, M. P., Anderton, C. R., Shapiro, J. P., Pamreddy, A., Winfree, S., Xiong, Y., He, Y., de Boer, I. H., Hodgin, J. B., Barisoni, L., Naik, A. S., Sharma, K., Sarwal, M. M., Zhang, K., Himmelfarb, J., Rovin, B., El-Achkar, T. M., Laszik, Z., He, J. C., Dagher, P. C., Valerius, M. T., Jain, S., Satlin, L. M., Troyanskaya, O. G., Kretzler, M., Iyengar, R., Azeloglu, E. U. & Kidney Precision Medicine Project. A reference tissue atlas for the human kidney. eng. *Science Advances* **8,** eabn4965. ISSN: 2375-2548 (June 2022).

152. THE HUMAN PROTEIN ATLAS. *B3GAT1 - Single cell type* 2022. `https://www.proteinatlas.org/ENSG00000109956-B3GAT1/summary/rna`.

153. Monaco, G., Lee, B., Xu, W., Mustafah, S., Hwang, Y. Y., Carré, C., Burdin, N., Visan, L., Ceccarelli, M., Poidinger, M., Zippelius, A., Pedro de Magalhães, J. & Larbi, A. RNA-Seq Signatures Normalized by mRNA Abundance Allow Absolute Deconvolution of Human Immune Cell Types. *Cell Reports* **26,** 1627–1640.e7. ISSN: 22111247. `https://linkinghub.elsevier.com/retrieve/pii/S2211124719300592` (Feb. 2019).

154. Can Ergen-Behr. *Personal communication with Can Ergen-Behr* 2021.

155. Andreatta, M., Corria-Osorio, J., Müller, S., Cubas, R., Coukos, G. & Carmona, S. J. Interpretation of T cell states from single-cell transcriptomics data using reference atlases. *Nature Communications* **12,** 2965. ISSN: 2041-1723. `https://www.nature.com/articles/s41467-021-23324-4` (May 2021).

156. Vargas-Rojas, M. I., Ramírez-Venegas, A., Limón-Camacho, L., Ochoa, L., Hernández-Zenteno, R. & Sansores, R. H. Increase of Th17 cells in peripheral blood of patients with chronic obstructive pulmonary disease. *Respiratory Medicine* **105,** 1648–1654. ISSN: 09546111. `https://linkinghub.elsevier.com/retrieve/pii/S0954611111001806` (Nov. 2011).

157. Sadeghi, A., Tahmasebi, S., Mahmood, A., Kuznetsova, M., Valizadeh, H., Taghizadieh, A., Nazemiyeh, M., Aghebati-Maleki, L., Jadidi-Niaragh, F., Abbaspour-Aghdam, S., Roshangar, L., Mikaeili, H. & Ahmadi, M. Th17 and Treg cells function in SARS-CoV2 patients compared with healthy controls. *Journal of Cellular Physiology* **236,** 2829–2839. ISSN: 0021-9541, 1097-4652. `https://onlinelibrary.wiley.com/doi/10.1002/jcp.30047` (Apr. 2020).

158. Zhang, W., Tian, X., Mumtahana, F., Jiao, J., Zhang, T., Croce, K. D., Ma, D., Kong, B. & Cui, B. The existence of Th22, pure Th17 and Th1 cells in CIN and Cervical Cancer along with their frequency variation in different stages of cervical cancer. *BMC Cancer* **15,** 717. ISSN: 1471-2407. `http://bmccancer.biomedcentral.com/articles/10.1186/s12885-015-1767-y` (Dec. 2015).

159. Luo, J., Zhang, M., Yan, B., Zhang, K., Chen, M. & Deng, S. Imbalance of Th17 and Treg in peripheral blood mononuclear cells of active tuberculosis patients. *The Brazilian Journal of Infectious Diseases* **21,** 155–161. ISSN: 14138670. `https://linkinghub.elsevier.com/retrieve/pii/S1413867016305852` (Mar. 2017).

160. Capone, A. & Volpe, E. Transcriptional regulators of T helper 17 cell differentiation in health and autoimmune diseases. *Frontiers in immunology* **11.** Publisher: Frontiers, 348 (2020).

161. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008,** P10008. ISSN: 1742-5468. `https://iopscience.iop.org/article/10.1088/1742-5468/2008/10/P10008` (Oct. 2008).

162. Meyer Zu Horste, G., Wu, C., Wang, C., Cong, L., Pawlak, M., Lee, Y., Elyaman, W., Xiao, S., Regev, A. & Kuchroo, V. *RBPJ controls development of pathogenic Th17 cells by regulating IL-23 receptor expression. Cell Rep 16 (2): 392–404* 2016.

163. De Biasi, S., Meschiari, M., Gibellini, L., Bellinazzi, C., Borella, R., Fidanza, L., Gozzi, L., Iannone, A., Lo Tartaro, D., Mattioli, M., Paolini, A., Menozzi, M., Milić, J., Franceschi, G., Fantini, R., Tonelli, R., Sita, M., Sarti, M., Trenti, T., Brugioni, L., Cicchetti, L., Facchinetti, F., Pietrangelo, A., Clini, E., Girardis, M., Guaraldi, G., Mussini, C. & Cossarizza, A. Marked T cell activation, senescence, exhaustion and skewing towards TH17 in patients with COVID-19 pneumonia. *Nature Communications* **11,** 3434. ISSN: 2041-1723. `https://www.nature.com/articles/s41467-020-17292-4` (July 2020).

164. Saraiva, D. P., Jacinto, A., Borralho, P., Braga, S. & Cabral, M. G. HLA-DR in cytotoxic T lymphocytes predicts breast cancer patients' response to neoadjuvant chemotherapy. *Frontiers in immunology.* Publisher: Frontiers, 2605 (2018).

165. Tippalagama, R., Singhania, A., Dubelko, P., Lindestam Arlehamn, C. S., Crinklaw, A., Pomaznoy, M., Seumois, G., deSilva, A. D., Premawansa, S., Vidanagama, D., Gunasena, B., Goonawardhana, N. D. S., Ariyaratne, D., Scriba, T. J., Gilman, R. H., Saito, M., Taplitz, R., Vijayanand, P., Sette, A., Peters, B. & Burel, J. G. HLA-DR Marks Recently Divided Antigen-Specific Effector CD4 T Cells in Active Tuberculosis Patients. *The Journal of Immunology* **207,** 523–533. ISSN: 0022-1767, 1550-6606. `http://www.jimmunol.org/lookup/doi/10.4049/jimmunol.2100011` (July 2021).

166. Machicote, A., Belén, S., Baz, P., Billordo, L. A. & Fainboim, L. Human CD8+ HLA-DR+ regulatory T cells, similarly to classical CD4+ Foxp3+ cells, suppress immune responses via PD-1/PD-L1 axis. *Frontiers in immunology.* Publisher: Frontiers, 2788 (2018).

167. Tu, A. A., Gierahn, T. M., Monian, B., Morgan, D. M., Mehta, N. K., Ruiter, B., Shreffler, W. G., Shalek, A. K. & Love, J. C. TCR sequencing paired with massively parallel 3′ RNA-seq reveals clonotypic T cell signatures. *Nature immunology* **20.** Publisher: Nature Publishing Group, 1692–1699 (2019).

168. Ly, L.-H. & Vingron, M. Effect of imputation on gene network reconstruction from single-cell RNA-seq data. *Patterns* **3,** 100414. ISSN: 26663899. `https://linkinghub.elsevier.com/retrieve/pii/S2666389921002889` (Feb. 2022).

169. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biology* **18,** 174. ISSN: 1474-760X. `http://genomebiology.biomedcentral.com/articles/10.1186/s13059-017-1305-0` (Dec. 2017).

170. Duda, J. *Gaussian AutoEncoder* arXiv:1811.04751 [cs, stat]. Jan. 2019. `http://arxiv.org/abs/1811.04751`.

171. LeCun, Y., Cortes, C. & Burges, C. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* **2** (2010).

172. Liu, Z., Luo, P., Wang, X. & Tang, X. *Deep learning face attributes in the wild* in *Proceedings of international conference on computer vision (ICCV)* (Dec. 2015).

173. Knop, S., Spurek, P., Tabor, J., Podolak, I., Mazur, M. & Jastrzebski, S. Cramer-wold auto-encoder. *Journal of Machine Learning Research* **21,** 1–28. `http://jmlr.org/papers/v21/19-560.html` (2020).

174. Ma, Q. & Xu, D. Deep learning shapes single-cell data analysis. *Nature Reviews Molecular Cell Biology* **23,** 303–304. ISSN: 1471-0072, 1471-0080. `https://www.nature.com/articles/s41580-022-00466-x` (May 2022).

175. Erfanian, N., Heydari, A. A., Iañez, P., Derakhshani, A., Ghasemigol, M., Farahpour, M., Nasseri, S., Safarpour, H. & Sahebkar, A. *Deep Learning Applications in Single-Cell Omics Data Analysis* preprint (Bioinformatics, Nov. 2021). `http://biorxiv.org/lookup/doi/10.1101/2021.11.26.470166`.

176. Yang, S., Corbett, S. E., Koga, Y., Wang, Z., Johnson, W. E., Yajima, M. & Campbell, J. D. Decontamination of ambient RNA in single-cell RNA-seq with DecontX. *Genome Biology* **21,** 57. ISSN: 1474-760X. `https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-1950-6` (Dec. 2020).

177. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* **11,** 3371–3408. http://jmlr.org/papers/v11/vincent10a.html (2010).

178. Young, M. D. & Behjati, S. SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data. *GigaScience* **9,** giaa151. ISSN: 2047-217X. https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giaa151/6049831 (Dec. 2020).

179. Ding, J., Smith, S. L., Orozco, G., Barton, A., Eyre, S. & Martin, P. Characterisation of CD4+ T-cell subtypes using single cell RNA sequencing and the impact of cell number and sequencing depth. *Scientific Reports* **10,** 19825. ISSN: 2045-2322. https://www.nature.com/articles/s41598-020-76972-9 (Nov. 2020).

180. Hausmann, F., Ergen-Behr, C., Khatri, R., Marouf, M., Hänzelmann, S., Gagliani, N., Huber, S., Machart, P. & Bonn, S. DiSCERN - Deep Single Cell Expression ReconstructioN for improved cell clustering and cell subtype and state detection. *bioRxiv,* 2022.03.09.483600. http://biorxiv.org/content/early/2022/11/01/2022.03.09.483600.abstract (Jan. 2022).

181. Martonik, D., Parfieniuk-Kowerda, A., Rogalska, M. & Flisiak, R. The Role of Th17 Response in COVID-19. *Cells* **10,** 1550. ISSN: 2073-4409. https://www.mdpi.com/2073-4409/10/6/1550 (June 2021).

182. Tang, C., Chen, S., Qian, H. & Huang, W. Interleukin-23: as a drug target for autoimmune inflammatory diseases. *Immunology* **135,** 112–124. ISSN: 0019-2805, 1365-2567. https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2567.2011.03522.x (Feb. 2012).

183. Rood, J. E., Maartens, A., Hupalowska, A., Teichmann, S. A. & Regev, A. Impact of the Human Cell Atlas on medicine. *Nature Medicine* **28,** 2486–2496. ISSN: 1078-8956, 1546-170X. https://www.nature.com/articles/s41591-022-02104-7 (Dec. 2022).

184. Arvanitidis, G., Hansen, L. K. & Hauberg, S. *Latent Space Oddity: on the Curvature of Deep Generative Models* arXiv:1710.11379 [stat]. Dec. 2021. http://arxiv.org/abs/1710.11379.

185. Merriam-Webster, Incorporated. *Definition of GEODESIC* 2023. https://www.merriam-webster.com/dictionary/geodesic.

186. Ding, J. & Regev, A. Deep generative model embedding of single-cell RNA-Seq profiles on hyperspheres and hyperbolic spaces. *Nature Communications* **12,** 2554. ISSN: 2041-1723. https://www.nature.com/articles/s41467-021-22851-4 (May 2021).

187. Lance, C., Luecken, M. D., Burkhardt, D. B., Cannoodt, R., Rautenstrauch, P., Laddach, A., Ubingazhibov, A., Cao, Z.-J., Deng, K., Khan, S., Liu, Q., Russkikh, N., Ryazantsev, G., Ohler, U., NeurIPS 2021 Multimodal data integration competition participants, Pisco, A. O., Bloom, J., Krishnaswamy, S. & Theis, F. J. *Multimodal single cell data integration challenge: results and lessons learned* preprint (Bioinformatics, Apr. 2022). http://biorxiv.org/lookup/doi/10.1101/2022.04.11.487796.

188. Efremova, M. & Teichmann, S. A. Computational methods for single-cell omics across modalities. *Nature Methods* **17,** 14–17. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/s41592-019-0692-4 (Jan. 2020).

189. Buenrostro, J. D., Wu, B., Litzenburger, U. M., Ruff, D., Gonzales, M. L., Snyder, M. P., Chang, H. Y. & Greenleaf, W. J. Single-cell chromatin accessibility reveals principles of regulatory variation. eng. *Nature* **523,** 486–490. ISSN: 1476-4687 (July 2015).

190. Marx, V. Method of the Year: spatially resolved transcriptomics. *Nature Methods* **18,** 9–14. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/s41592-020-01033-y (Jan. 2021).

191. Zhu, C., Preissl, S. & Ren, B. Single-cell multimodal omics: the power of many. *Nature Methods* **17,** 11–14. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/s41592-019-0691-5 (Jan. 2020).

192. Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K. L., Streets, A. & Yosef, N. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature Methods* **18,** 272–282. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/s41592-020-01050-x (Mar. 2021).

193. Yang, K. D., Belyaeva, A., Venkatachalapathy, S., Damodaran, K., Katcoff, A., Radhakrishnan, A., Shivashankar, G. V. & Uhler, C. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *Na-*

ture Communications **12,** 31. ISSN: 2041-1723. https://www.nature.com/articles/s41467-020-20249-2 (Jan. 2021).

194. Perkel, J. M. Single-cell proteomics takes centre stage. *Nature* **597,** 580–582. ISSN: 0028-0836, 1476-4687. https://www.nature.com/articles/d41586-021-02530-6 (Sept. 2021).

195. Čuklina, J., Lee, C. H., Williams, E. G., Sajic, T., Collins, B. C., Rodríguez Martínez, M., Sharma, V. S., Wendt, F., Goetze, S., Keele, G. R., Wollscheid, B., Aebersold, R. & Pedrioli, P. G. A. Diagnostics and correction of batch effects in large-scale proteomic studies: a tutorial. *Molecular Systems Biology* **17.** ISSN: 1744-4292, 1744-4292. https://onlinelibrary.wiley.com/doi/10.15252/msb.202110240 (Aug. 2021).

196. Lazar, C., Gatto, L., Ferro, M., Bruley, C. & Burger, T. Accounting for the Multiple Natures of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation Strategies. *Journal of Proteome Research* **15,** 1116–1125. ISSN: 1535-3893, 1535-3907. https://pubs.acs.org/doi/10.1021/acs.jproteome.5b00981 (Apr. 2016).

A Datasets

Table 10: *Overview of all single cell and bulk sequencing datasets used in this study.* The table shows the dataset name, size of the dataset, the sequencing technology, cell types as annotated in the original study and a hyperlink to the publication.

Dataset	*Method*	*Cell Types*	*Publication or Download link*
pancreas (8 569 cells)	SMARTSeq2, Fluidigm C1, CelSeq, CelSeq2, inDrops	alpha, beta, ductal, acinar, delta, gamma, activated_stellate, endothelial, quiescent_stellate, macrophage, mast, epsilon, schwann	[135]
difftec (31 021 cells)	10x Chromium v2, 10x Chromium v3, SMARTSeq2, Seq-Well, inDrops, Drop-seq, CelSeq2	Cytotoxic T cell, CD4$^+$ T cell, CD14$^+$ monocyte, B cell, Natural killer cell, Megakaryocyte, CD16$^+$ monocyte, Dendritic cell, Plasmacytoid dendritic cell, Unassigned	[5]

Table 10: *Overview of all single cell and bulk sequencing datasets used in this study continued.*

Dataset	***Method***	***Cell Types***	***Publication or Download link***

snRNA-seq & scRNA-seq (12 423 cells)	snRNA-seq and scRNA-seq using Chromium single-cell 3′ v3	Epithelial cells, Macrophages, Hepatocytes, T cells, Endothelial cell, Fibroblasts, B cells, NK cells	`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM4186980`
			`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM4186974`

Table 10: *Overview of all single cell and bulk sequencing datasets used in this study continued.*

Dataset	***Method***	***Cell Types***	***Publication or Download link***
covid-lung (56 645 cells)	10X Chromium Single Cell V(D)J Reagent Kit v1.1	CD8 T, TREG, CD4_CD8 proliferating, B cell, CD4_TCM, TRM1, TR1, CD8_TCM, T senescent, CD8_TEM, TEM17, T antiviral, alveolar MΦ, TRM17, M1, CD4_CD8 stressed TCM, CD4_TSCM, MAIT, Innate like, Neutrophils, doublets, CD4_CD8 lnc rich, aged Neutrophils, M1 HSP$^+$, Mast, DC, M1 Mono-derived, M2 profibrotic, Epithelial, Neutrophil, Macrophage	[140]
covid-blood (83 709 cells)	10X Chromium Single Cell V(D)J Reagent Kit v1.1	CD3$^+$ cells	[140]

Table 10: *Overview of all single cell and bulk sequencing datasets used in this study continued.*

Dataset	***Method***	***Cell Types***	***Publication or Download link***

citeseq
(6 592 cells)
	10x Genomics Single Cell and CITE-seq	B cells, CD4 T cells, NK cells, CD14^{+} Monocytes, FCGR3A^{+} Monocytes, CD8 T cells	`https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100866`
			`https://github.com/YosefLab/scVI-data/raw/master/pbmc_metadata.pickle`

Table 10: *Overview of all single cell and bulk sequencing datasets used in this study continued.*

Dataset	*Method*	*Cell Types*	*Publication or Download link*
bulk (9852 cells)	SMART-seq v4	Naive CD4, Memory CD4, TH1, TH2, TH17, Tfh, Fr. I nTreg, Fr. II eTreg, Fr. III T, Naive CD8, Memory CD8, CM CD8, EM CD8, TEMRA CD8, NK, Naive B, USM B, SM B, Plasmablast, DN B, CL Monocytes, Int Monocytes, NC Monocytes, mDC, pDC, Neutrophils, LDG	[138]

Table 10: *Overview of all single cell and bulk sequencing datasets used in this study continued.*

Dataset	*Method*	*Cell Types*	*Publication or Download link*
Kidney snRNA-seq & scRNA-seq (82 701 cells)	10x Genomics Chromium v3	None (not annotated)	`https://atlas.kpmp.org/repository/?facetTab=participants` Patients: 3010018, 3010034, 3210003, 3210034, 3310005, 3310006, 3410050, 3410184, 3410187

Table 11: *Detailed quality and batch information for all single cell and bulk sequencing datasets used in this study.* For each batch, the number of cells, the mean number of counts per cell, and the mean number of expressed genes per cell are listed. For the difftec dataset, the batch names were slightly adjusted. Their published batch names are written in brackets.

Dataset	**Batch**	**Number of cells**	**Mean number of counts per cell**	**Mean number of genes**
pancreas	smartseq2	2394	451021.4	6214.0
	fluidigmc1	638	1580155.4	8127.4
	celseq	2285	11161.1	3466.8
	celseq2	1004	23394.2	5274.9
	indrop	8569	5828.2	1887.2
difftec	dropseq (pbmc1_Drop-seq)	3222	1282.0	676.0
	indrops (pbmc1_inDrops)	3222	566.3	362.4
	seqwell (pmbc1_Seq-Well)	3222	1035.3	567.2
	chromium-v3 (pbmc1_10x Chromium (v3))	3222	4891.3	1514.1
	chromium-v2 (pbmc1_10x Chromium (v2) A)	3222	2120.0	795.4
	chromium-v2B (pmbc1_10x Chromium (v2) B)	3222	2512.4	870.8
	smartseq2 (pbmc1_Smart-seq2)	253	385914.3	2434.6
	celseq2 (pbmc1_CEL-Seq2)	253	6057.3	2585.4
	dropseq-2 (pbmc2_Drop-seq)	3362	2141.0	977.7
	seqwell-2 (pbmc2_Seq-Well)	551	692.6	421.8

Table 11: *Detailed quality and batch information for all single cell and bulk sequencing datasets used in this study continued.*

Dataset	Batch	Number of cells	Mean number of counts per cell	Mean number of genes
	smartseq2-2 (pbmc2_Smart-seq2)	273	292924.3	2795.4
	celseq2-2 (pbmc2_CEL-Seq2)	273	5949.3	2556.6
	chromium-v2-2 (pbmc2_10x Chromium (v2))	3362	2860.7	1131.4
	indrops-2 (pbmc2_inDrops)	3362	1249.5	619.5
snRNA-seq & scRNA-seq	sn-lq	7260	2206.6	1308.7
	sc-hq	5163	4634.5	1214.6
covid-lung	Bacterial	14591	9627.2	1617.4
	SARS-CoV-2	42054	10284.4	1719.5
covid-blood	Bacterial	22199	5861.6	1703.0
	SARS-CoV-2	61510	5388.6	1700.7
citeseq	citeseq	6592	1391.8	797.8
bulk	bulk	9852	881440.6	13103.8
Kidney snRNA-seq & scRNA-seq	kidney-lq (snRNA-seq)	52934	6532.8	2462.7
	kidney-hq (snRNA-seq)	29767	4449.6	1546.0

Acronyms

AMI adjusted mutual information. 16, 52, 56, 58–60, 108, 158, 159, 171

ARI adjusted Rand index. 15, 16, 52, 56, 57, 59, 60, 108, 158, 159, 171

AUROC area under the receiver operating characteristic curve. 37

BAL bronchoalveolar lavage. 44, 114

BN batch normalization. 21–24

CBN conditional batch normalization. 24

CCA Canonical Correlation Analysis. 10

CIN conditional instance normalization. 21, 22, 24, 31, 32, 82, 106, 108, 115, II, IV

CLN conditional layer normalization. 24, 34, 35

COVID-19 coronavirus disease 2019. 82, 97, 101, 103, 104, 114, 115, 170, III, V

DEA differential expression analysis. 12, 13, 66, 69–73, 75–77, 81, 108, 109, 162–164, 172

DEG differential expressed gene. 63–65, 71, 163, 171

ELBO Evidence lower bound. 18

FACS Fluorescence Activated Cell Sorting. 43, 93, 96, 97, 99, 113, 116, 168

GAN generative adversarial network. 17, 19, 26, 155

KEGG Kyoto Encyclopedia of Genes and Genomes. 13

LN layer normalization. 21, 23, 24, 34

MMD Maximum Mean Discrepancy. 20, 33, 39, 45, 51, 156, 157

MSE mean squared error. 12, 18

PBMC peripheral blood mononuclear cell. 82, 93, 94, 97, 99, 113, 114, 116, 168

PCA Principle Component Analysis. 9, 10, 14, 52, 54, 59, 79, 88, 96, 164, 167

PCR polymerase chain reaction. 3

RKHS reproducing kernel Hilbert space. 20

RNA-seq RNA-sequencing. 3, 6, 8–10, 93, 97, 106, 113–116, II, V

scRNA-seq single-cell RNA-sequencing. 3–14, 17, 19, 21, 22, 24, 25, 32–34, 36, 37, 40–44, 49, 50, 52, 54, 56, 59, 84, 85, 88, 91–94, 96, 97, 101, 105–107, 110, 112–118, 167, 171, II, IV, V

snRNA-seq single-nuclei RNA-sequencing. 5, 42, 43, 52, 53, 55–57, 59–65, 72, 84, 88, 91, 92, 107, 110, 112, 116, 157–160, 167, 171, II, V

SVD Singular Value Decomposition. 10

t-SNE t-distributed stochastic neighbor embedding. 13, 14, 52–54, 79, 83, 90–93, 95, 96, 98, 100, 112, 157, 158, 164–168

TCR T cell receptor. 44, 84–86, 101, 102, 165, 169

UMAP Uniform Manifold Approximation and Projection. 13, 14

UMI unique molecular identifier. 3, 5

VAE variational autoencoder. 10, 11, 18–21, 28, 29, 36, 109, 110, 117, 155, 156

WAE Wasserstein autoencoder. 17, 19–21, 29, 32, 36, 106, 109, 110, 116, 117, 156, II, IV

ZINB Zero-inflated negative Binomial. 7, 11, 33

List of Figures

List of Tables

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, 26. Januar 2022 F. Ha
Ort, Datum Unterschrift