Preprocessing and Downstream Analysis of scATAC-seq Data

Pia Baronetzky Markus Franke Niklas Kemper George Tsitsiridis

Abstract

In recent years, numerous normalization techniques have been developed for scRNA-seq data, improving upon separating technical variance from biological variance. We analyse and benchmark whether and how these methods can be applied to scATAC-seq and show that the developement of residual transformations similar to SCTransform, but specific to scATAC-seq data, promise the potential to provide better cell type differentiation than simple library size normalization. The method SCTransform performs best on our dataset using metrics assessing the clustering quality after normalization.

1. Introduction

The development of experimental techniques to study chromatin accessibility in single cells has led to the need for suitable methods to analyze this type of data [2]. Normalization methods have been developed to improve the ability to differentiate biological from technical variance in scRNA-seq data. Normalization is an important step in the analysis of scRNA-seq data with a number of algorithms (including SCRAN and SCTransform) developed for and and analyzed with scRNA-seq data. However, scATAC-seq data is much sparser compared to scRNA-seq data. Additionally, scATAC-seq data is often binarized before analysis. Because of these differences in input data, it is not clear if normalization methods designed for scRNA-seq data will perform equally well on scATAC-seq data.

After normalization, features that show high variance should also be differentially accessible (DA). Assessing the effect of normalization in downstream DA analysis though requires experimentally known differential features. As an alternative simulated datasets were used to artificially create DA and assess the effect of normalization methods in DA analysis.

2. Results

For the evaluation of the normalization methods, a mouse brain dataset was used which contained cell type annotations. Brain data sets are broadly accessible as brain samples can be preserved well. Moreover, the brain contains diverse, fully differentiated cell types, which makes it easier to evaluate cell type separation after normalization.

2.1. The Fragment Counts follow a Poisson distribution with Overdispersion

It has been shown that scATAC-seq data contains information that extends beyond a simple binary measure of accessibility and that the fragment count can roughly be approximated using the Poisson distribution.[9]
Indeed, for the mouse dataset used in this report, the fragment count follows an overdispersed Poisson distribution as visualized in figure 1.

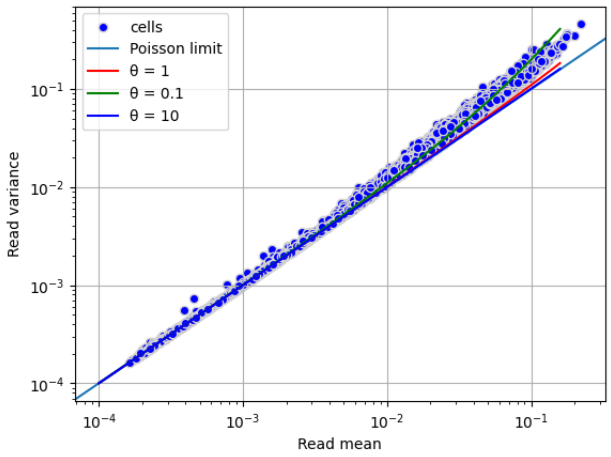

Figure 1. Variance of fragment counts grows approximately quadratically to the fragment count mean, consistent with a negative binomial distribution with overdispersion parameter $\theta = 0.1$.

2.2. Unnormalized data

Figure 7, 8 and 9 show UMAPs of our dataset before normalization, colored by library size, leiden clusters and cell types. In the UMAPs, clusters are driven by library size to a large extent, do not match cell types very well and tend to be elongated. It can also be seen that library size plays a role in figure 10 which shows the high correlation of library size to the first principal component in PCA (Here, the correlation is 0.98). This shows that without any library size normalization technique the technical variation in the data decreases

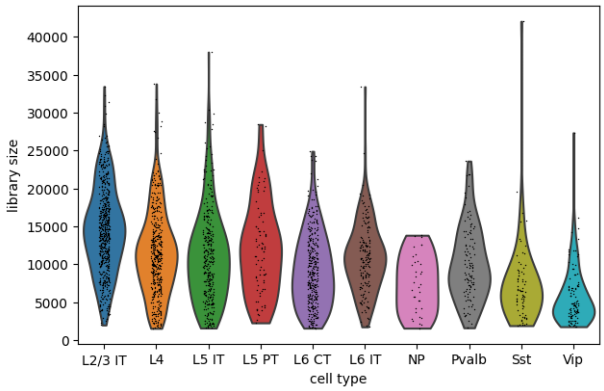

Figure 2. Violin plots of the library size grouped by cell types for our dataset.

the quality of the subsequent analysis steps (e.g. clustering). This underlines the importance of adequate normalization techniques for scATAC-seq data.

2.3. Library Size Based Normalization Techniques

The aim of a normalization technique is to remove the differences in library size caused by technical variation. However, different cell types can have different amounts of open chromatin and, hence, there can also be differences in library size caused by biological variation as can be seen in figure 2. Hence, normalization techniques have to try to preserve the biological variation while removing all technical variation. Simple library size normalization presumes that every cell has the same biological library size and that variations in library size are purely caused by technical effects. Even tough this presumption is wrong, simple library size normalization followed by a log transformation greatly improves clustering compared to the unnormalized data as can be seen in figure 17.

Library size normalisation followed by a log transformation and another library size normalisation as proposed by [1] as well as regressing out the first principal component also improve the clustering compared to the unnormalized data. However, figure 17 shows that the clusters are less fine grained and cannot separate all cell types.

2.4. SCRAN

SCRAN, a method working with a pooling-deconvolution approach, performs worse on our scATAC-seq data in comparison with other methods like SCTransform, even though it is thought particularly for very sparse data. This can can be seen in figure 3. SCRAN performed best with leiden clusters, as can be seen in figure 11 and 12, and with a value `min.mean` of 0.1.

2.5. Residual Transformations

The idea behind residual transformations is to construct generalized linear models for each feature separately with the UMI counts as dependent variable and the openness as explanatory variable. This relies on the assumption that the fitted values based on UMI counts represent technical/sequencing variation. We then define the matrix of the residuals as our new normalized data, with the consequence of highlighting features according to the evidence that they are non-uniformly expressed.
The biggest challenge in adapting such a model for scATAC data is its sparsity and limited sensitivity, with scATAC-seq detecting only 5%–15% of accessible regions[11]. Due to the low probability that biologically accessible regions actually get sequenced, almost all fitted values are close to 0 as can be seen reflected in the residuals in figure 14 for binary data and figures 15, 16 for count data.
Furthermore, when using fixed overdispersion parameters in both the binary case and SCTransform, we encountered slopes with low absolute values and high uncertainty, together with an unusually high amount of negative slopes in the logistic regression model specifically (see Figure 6). This means that for these features a higher UMI count leads to a lower probability of observing an open frame in the model. This is an overfit, as it is not driven by sequencing variation, but biological variation, and as such we are removing part of the biological variation instead of magnifying this feature due to the evidence that it is non-uniformly expressed.

2.6. Metrics on annotated data set

To measure the quality of the different normalization techniques that we analyzed, we used four different metrics. The two library size correlation metrics (globally and per cell type) try to measure the removal of technical variation. The average silhouette width and the adjusted rand index quantify the conservation of biological information. Ideally, a normalization technique would remove all technical variation while retaining all biological variation. The results of the metrics can be found in figure 3.

Three different versions of SCTransform were tested (see methods 4.4), where the best performing residual transformation approach was SCTransform with a fixed overdispersion parameter $\theta = 0.1$ and no kernel regression. SCTransform on count data outperformed the binary residual transformation significantly in regards to cell-type classification, regardless of method, while the binary residual transformation significantly outperforms unnormalized data.
Overall, the metrics show that SCTransform performs best in regards to biological conservation on our data set. Additionally, SCTransform also performs well on the library size

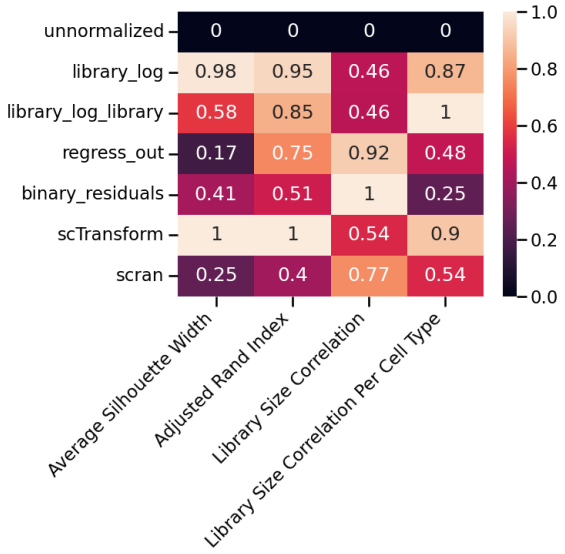

Figure 3. Results of the metrics on the mouse brain data set for unnormalized data, library size normalisation followed by log transformation, library size normalisation followed by log transformation and another library size normalisation, regress out normalisation, binary residual normalisation, SCTransform and scran. The results are scaled such that 1 is assigned to the best performing method and 0 to the worst performing method.

correlation per cell type metric. However, in all metrics SCTransform is closely followed by the much simpler library size normalization method. All other methods perform considerably worse on the biological conservation metrics. As expected, the unnormalized data achieves the worst results on all metrics.

2.7. Differential accessibility analysis on simulated data

We run DA analysis on different DA configurations as defined on Table 1. Figure 4 shows the F1 score for configurations with high *DA level* (see Methods). The F1 score is the harmonic mean of the precision and recall. The performance of each normalization method is dependent on the *library size level* (see Methods). For higher library size *SCTransform* and *library_log_library* perform significantly better than all other methods. Higher library size effectively means higher variance which means that those methods are more effective in stabilizing variance. For lower library size though *scran* and *library_log* perform significantly better than all other methods. Furthermore, library size normalization on binary data performs worse than all other normalizations on raw counts. The performance on other DA configurations can be seen in Figures 13 and 18. The significance was assessed using 95% confidence intervals.

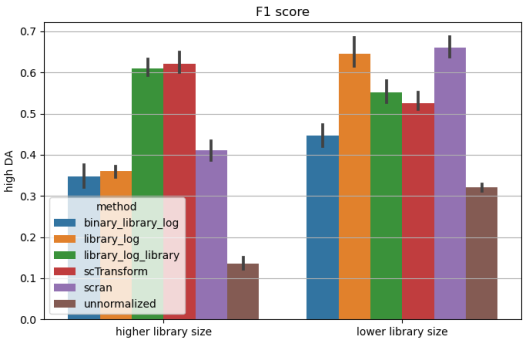

Figure 4. Barplot showing the F1 score for DA analysis of different normalization methods for high *DA level* (Benjamini-Hochberg adjusted p-values at FDR = 0.05). The error bars show the 95% confidence intervals.

3. Discussion

Simple library size normalization achieved surprisingly good results on the metrics for the annotated data set. This might indicate that simple library size normalization is sufficient for many use cases.

However, SCTransform outperformed simple library size normalization, despite not yet being refined for scATAC-seq data. Given the differences in overdispersion among datasets, with some even showing little to no overdispersion at all[9], the optimal handling of overdispersion in scATAC-seq data remains unclear.
We also believe that further research into residual transformation methods using binarized counts would be of value, where possible avenues of fixing the overfits in the binary residual model include employing the celltypes as random effects in the logistic regression, using a beta binomial logistic regression model to account for overdispersion or adding an additional step of kernel regression to smooth the parameters along the UMI.
On the other hand, the fact that SCTransform without kernel regression and with just a fixed overdispersion parameter performed best, despite also suffering from very low absolute values for the slopes compared to SCTransform with kernel regression and free overdispersion parameter θ, implies that the count data improved the procedure for this specific data set.

For DA analysis, the normalization method of choice strongly depends on the library size of the compared cells and the DA level of the differentiated features. There is no method that performs best in all possible DA configurations. However, finding the most biologically meaningful configuration would be a step towards the right direction.

4. Methods

Before normalization, we performed quality control following the Episcanpy tutorial by Patrick Hanel[4].

4.1. Simple Library Size Normalization

Simple library size normalisation is one of the most common approaches both for normalizing the library size of scATAC-seq as well as scRNA-seq data. All counts $x_{i,g}$ of cell i for gene g are transformed by

$$x'_{i,g} = s \cdot \frac{x_{i,g}}{\sum_g x_{i,g}} \tag{1}$$

such that every cell has the same sum of counts s after the transformation. For the value of s we have used the median library size. The scaling to the median library size size is followed by a log1p transformation for variance stabilization.

For scRNA-seq it was proposed to apply another scaling to the same library size after the log transformation in [1]. This is also applied to scATAC-seq here.

4.2. Regress Out Normalization

Another simple normalization approach is to regress out the first principal component. For computing the first principal component and regressing it out, we used the corresponding episcanpy methods [4].

4.3. SCRAN

SCRAN is a normalization method thought particularly for RNA data that addresses the problem that most normalization methods are hard to put into practice for noisy single cell data with many zeros [8]. It is known for preserving biological variation.

Its approach consists of a pooling and a deconvolution step. First, pools of cells are created and the expression values for the cells in each pool are summed together. These pooled expression values are normalized against an average reference pseudo-cell [8].

A size factor for a pool is defined as the median ratio between the count sums and the average across all genes. The size factor for the pool can be expressed as a linear equation of the size factors for the cells. To obtain size factors for single cells, this can be repeated for multiple pools, such that the linear system of equations can be solved [8].

We conducted experiments with the scran parameters `clusters` and `min.mean`. When giving a clustering, cells within these clusters are sorted by increasing library size and a sliding window is applied to this. Each window is a pool of cells with similar library sizes. This prevents cells with very small library size to be pooled together with cells with a large library size. We tested given no clusters as `clusters` parameter, giving leiden clusters and cell type clusters. The parameter `min.mean` is responsible for filtering out genes that do not occur often, we tested values between 0.0 and 0.6.

4.4. Residual Transformations

4.4.1. The Binary Case

We fit a simple logistic regression with UMI counts as dependent variable and openness as explanatory variable:

$$\begin{aligned} \mu_{ij} &= \frac{1}{1+^{(\beta_{0_i}+\beta_{1_i} m_j)}} \\ z_{ij} &= x_{ij} - \mu_{ij} \end{aligned} \tag{2}$$

where μ_{ij} is the expected fitted value in the logistic regression model, β_{0_i} and β_{1_i} are the model parameters, $m_j = \sum_i x_{ij}$ is the vector of the amount of open features for each cell j, x_{ij} is the observed openness of feature i in cell j and z_{ij} is the residual of feature i in cell j.

4.4.2. Using Fragment Counts

SCTransform fits a regularized negative binomial regression with UMI counts as dependent variable and fragment count as explanatory variable:

$$\begin{aligned} \mu_{ij} &= \exp(\beta_{0_i} + \beta_{1_i} \log_{10} m_j) \\ z_{ij} &= \frac{x_{ij} - \mu_{ij}}{\sigma_{ij}} \\ \sigma_{ij} &= \sqrt{\mu_{ij} + \frac{\mu_{ij}^2}{\theta_i}} \end{aligned} \tag{3}$$

where μ_{ij} is the expected UMI count in the regularized NB regression model, β_{0_i} and β_{1_i} are the model parameters, $m_j = \sum_i x_{ij}$ is the vector of the sum of fragments for each cell j, x_{ij} is the observed fragment count of feature i in cell j and z_{ij} is the Pearson residual of feature i in cell j.

Parameter estimates are then smoothed using kernel regression, and the residuals are recalculated with the new model parameters after the smoothing.
A recent paper has critised this approach, claiming that estimating overdispersion for each gene in scRNA-seq is an overfit and that setting a fixed overdispersion parameter for all genes without kernel regression is a more sensible approach [6].
However S. Choudhary and R. Satija have published a paper in response to that, detailing evidence that the degree of overdispersion varies not only widely across data sets and gene abundances, but also among biological systems, motivating their use for gene-level overdispersion parameters in scRNA-seq[3].
The same paper also lead to an updated version of SCTrans-

form, called SCTransform v2, which constitutes the third method tested in this report.

4.5. Metrics on annotated data set

4.5.1. Removal of technical variation

An important aim of any normalization technique is to remove all unwanted technical variation. To quantify this effect, we computed the absolute value of the Pearson correlation between the library size and the first principal component. We calculated this absolute correlation per cell type (and averaged over all cell types) and per all cells.

4.5.2. Conservation of biological information

To asses the performace of the normalization methods in regards of conservation of biological information, we compared the leiden clusters after normalization with the give cell type labels. For this, we used as metrics the *Adjusted Rand Index* which classifies how similar the leiden clusters (of optimal resolution) are to the true cell type classes and the *Average Silhouette Width* which measures how densly the cells of the same label are clustered together. We used the implemntations in the SCIB software package [7].

4.6. Differential accessibility analysis

To assess the effect of different normalization methods in DA analysis we used the following approach. First, we used the cell type annotation to create 10 different subsets of the original filtered mouse brain dataset. For each cell type we simulated a new dataset according to 4 different DA configurations as defined on Table 1. Next we performed different normalizations on each simulated dataset and performed DA analysis on each. We assessed the performance of each normalization method on each cell type and DA scenario by calculating several classification metrics. Finally, we merged the results of all cell types by computing 95% confidence intervals for each metric. The final output of the analysis was several classification metrics per normalization method per DA scenario. More detailed information about each step is given below.

4.6.1. Data simulation

The R package *simATAC* [10] was used to perform the data simulations. SimATAC simulates new datasets by first estimating model parameters on a given dataset and then sampling new parameters from the estimated model distributions. To create artificial DA on different levels we subsequently adjust some of those estimated parameters according to the given DA configuration.

Each DA configuration consists of 2 arrays called *cell-groups* and *feature-groups*. Each cell-group represents a group of cells that have similar biological properties (e.g. cells from the same cell type). Each feature-group represents features that have similar counts in each cell group (e.g. regions that are co-regulated by the same transcription factors). Each cell-group:feature-group pair has certain properties *library size* and *score*. Library size represents the library size of the given cell as defined by simATAC. Score represents the *non-zero cell proportion* (i.e. accessibility) of the given feature as defined by simATAC. By adjusting the score of a specific feature-group accross different cell-groups one could create DA feature-groups of the desired level. Similarly, by adjusting the library size of 2 cell-groups one could mask the DA between them.

In total, we defined 4 different DA configurations. In each configuration there is a *control* cell-group and a *differentiated* cell-group. The control cell-group is constant for all DA configurations whereas the differentiated cell-group always contains feature-groups with higher counts. Each configuration captures 2 properties, *DA level* and *library size level*. DA level refers to the degree of change between control and differentiated cell-group. Library size level refers to the library size of the differentiated cell-group and can be either higher or lower than the control. These are summarised on Table 1.

DA level	**Library size level**
low	lower than control
low	higher than control
high	lower than control
high	higher that control

Table 1. DA configurations

4.6.2. Test method

To perform DA tests we used episcanpy's built-in functions using a t-test as test method. The result is a list of ranked features with their respective p-values. The p-values were adjusted using the Benjamini-Hochberg procedure with an FDR cutoff of 0.05.

4.6.3. Classification metrics

DA analysis assessment was performed using the following classification metrics.

$$\begin{aligned} \text{Precision} = \text{Prec} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ F_1 &= \frac{2 * \text{Prec} * \text{Recall}}{\text{Prec} + \text{Recall}} = \frac{2 * TP}{2 * TP + FP + FN} \end{aligned} \tag{4}$$

References

[1] Booeshaghi, A. S., Hallgrímsdóttir, I. B., Gálvez-Merchán, Á., and Pachter, L. Depth normalization for single-cell genomics count data. *bioRxiv*, pp. 2022–05, 2022.

[2] Buenrostro, J., Wu, B., Chang, H., and Greenleaf, W. Atac-seq: A method for assaying chromatin accessibility genome-wide. In *Curr Protoc Mol Biol*, 2015. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4374986/.

[3] Choudhary, S. and Satija, R. Comparison and evaluation of statistical error models for scRNA-seq. *Genome Biology*, 23(1):27, January 2022. ISSN 1474-760X. doi: 10.1186/s13059-021-02584-9.

[4] Danese, A., Richter, M. L., Fischer, D. S., Theis, F. J., and Colomé-Tatché, M. EpiScanpy: Integrated single-cell epigenomic analysis. Preprint, Bioinformatics, May 2019.

[5] Hafemeister, C. and Satija, R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biol*, 20(1):296, December 2019. ISSN 1474-760X. doi: 10.1186/s13059-019-1874-1.

[6] Lause, J., Berens, P., and Kobak, D. Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. *Genome Biol*, 22(1):258, December 2021. ISSN 1474-760X. doi: 10.1186/s13059-021-02451-7.

[7] Luecken, M. D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Mueller, M. F., Strobl, D. C., Zappia, L., Dugas, M., Colomé-Tatché, M., and Theis, F. J. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022. doi: 10.1038/s41592-021-01336-8. URL https://doi.org/10.1038/s41592-021-01336-8.

[8] Lun, A. T. L., Bach, K., and Marioni, J. C. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. In *Genome Biology*, 2016. URL https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0947-7.

[9] Martens, L. D., Fischer, D. S., Theis, F. J., and Gagneur, J. Modeling fragment counts improves single-cell ATAC-seq analysis, May 2022.

[10] Navidi, Z., Zhang, L., and Wang, B. simATAC: A single-cell ATAC-seq simulation framework. *Genome Biol*, 22(1):74, December 2021. ISSN 1474-760X. doi: 10.1186/s13059-021-02270-w.

[11] Preissl, S., Fang, R., Huang, H., Zhao, Y., Raviram, R., Gorkin, D. U., Zhang, Y., Sos, B. C., Afzal, V., Dickel, D. E., Kuan, S., Visel, A., Pennacchio, L. A., Zhang, K., and Ren, B. Single-nucleus analysis of accessible chromatin in developing mouse forebrain reveals cell-type-specific transcriptional regulation. *Nat Neurosci*, 21(3):432–439, March 2018. ISSN 1097-6256, 1546-1726. doi: 10.1038/s41593-018-0079-3.

A. Appendix

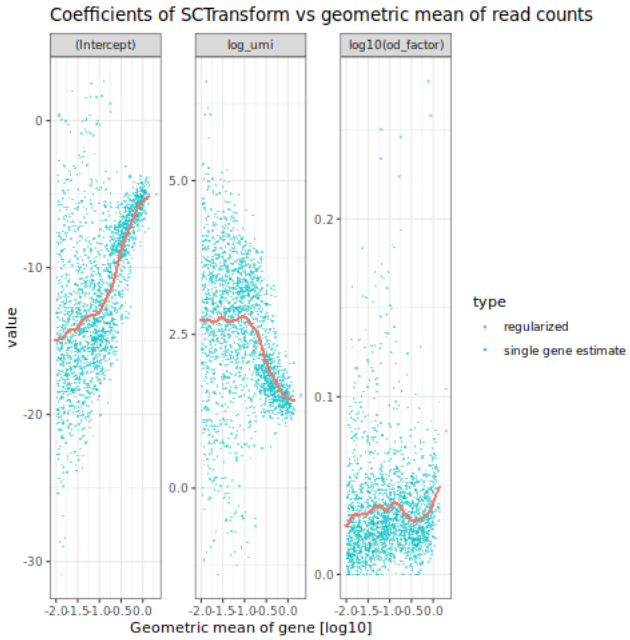

Figure 5. Plot of the Coefficients of SCTransform with variable θ and kernel regression. Note that the log slopes are mostly larger than 0.

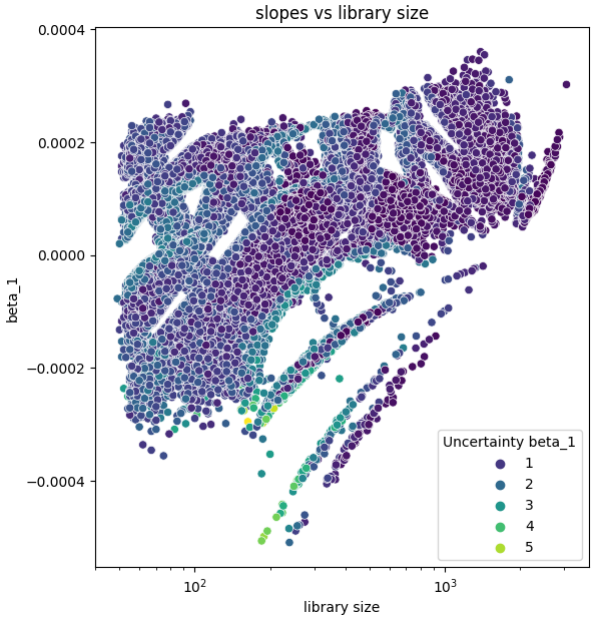

Figure 6. The slopes β_{1_i} of the logistic regression models. Uncertainty was calculated by bootstrapping the parameter estimation, we sampled from all cells with replacement 15 times, to fit 15 models, and used the standard deviation of parameter estimates across the 15 bootstraps divided by the standard deviation of the bootsrap-mean value across all genes (similarly to the procedure in [5]). Values greater than one indicate high uncertainty.

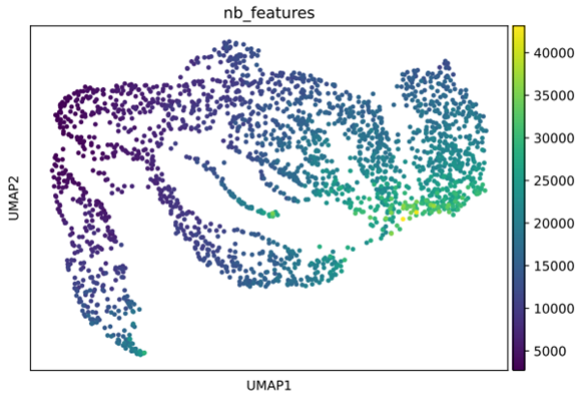

Figure 7. UMAP of the mouse brain dataset colored by library size.

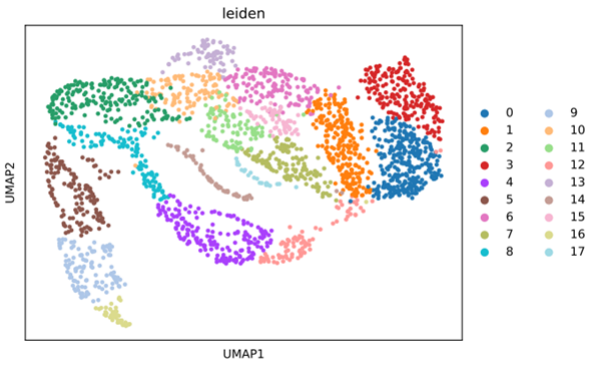

Figure 8. UMAP of the mouse brain dataset colored by leiden clusters.

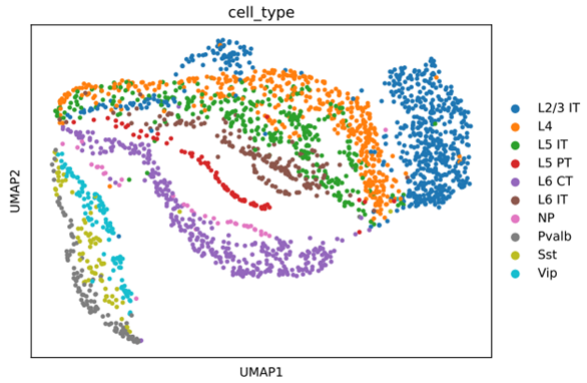

Figure 9. UMAP of the mouse brain dataset colored by cell types.

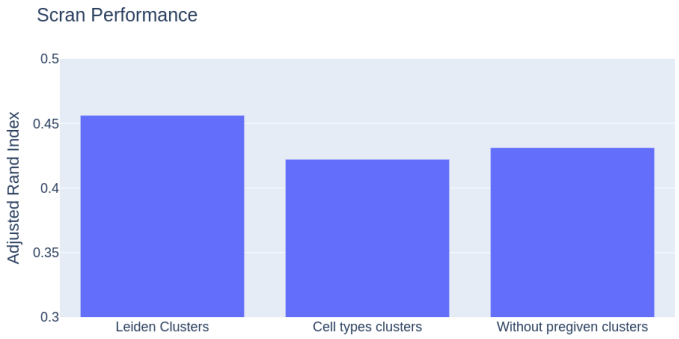

Figure 12. Adjusted Rand Index of SCRAN with different clustering.

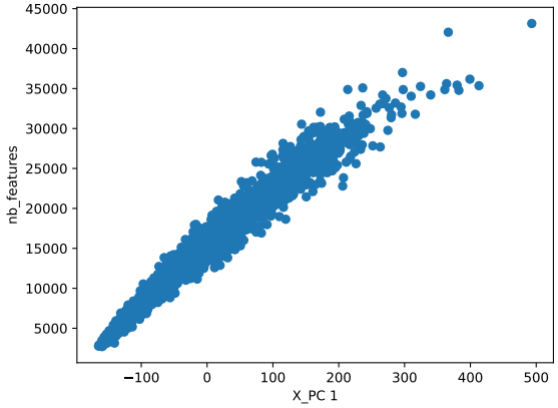

Figure 10. Library size/ First principal component correlation.

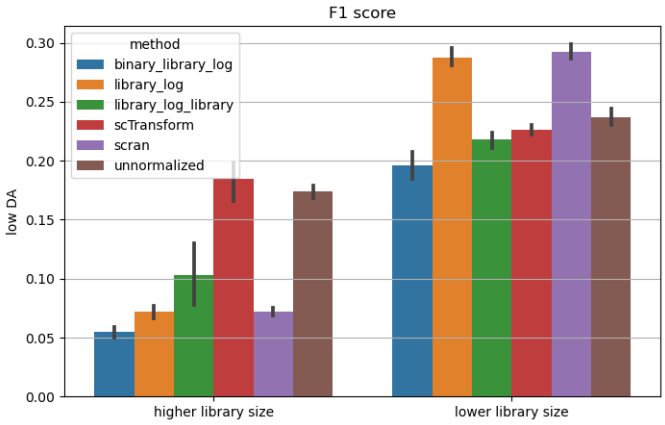

Figure 13. Barplot showing the F1 score for DA analysis of different normalization methods for low *DA level*.

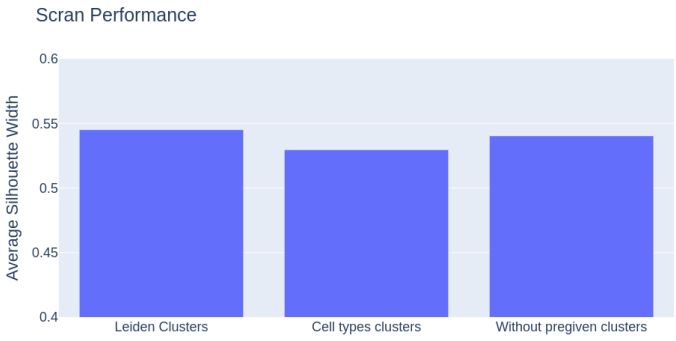

Figure 11. Average Silhouette Score of SCRAN with different clustering.

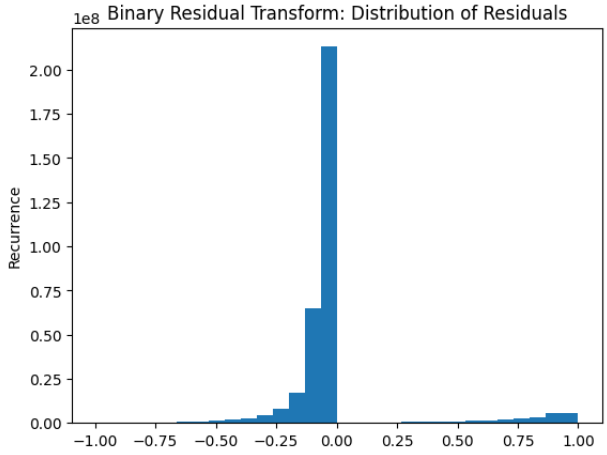

Figure 14. Distribution of residuals, i.e. the values of the normalized matrix after binary residual transformation

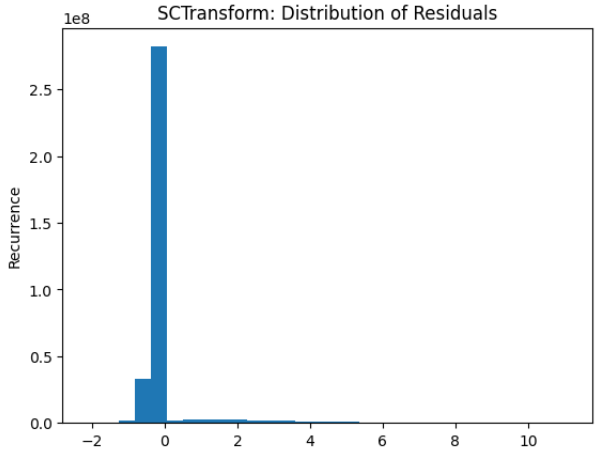

Figure 15. Distribution of residuals, i.e. the values of the normalized matrix after SCTransform

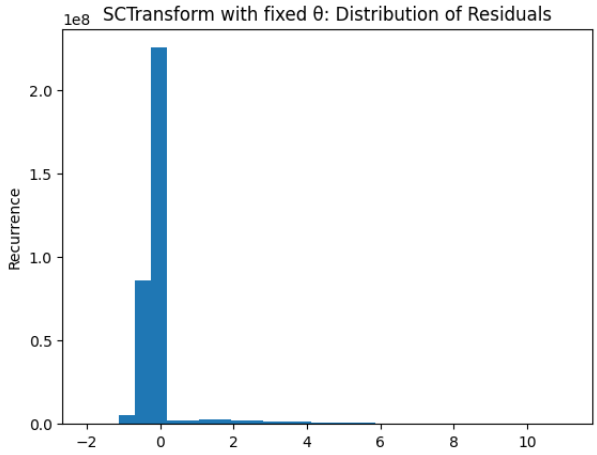

Figure 16. Distribution of residuals, i.e. the values of the normalized matrix after SCTransform with fixed overdispersion parameter $\theta = 0.1$

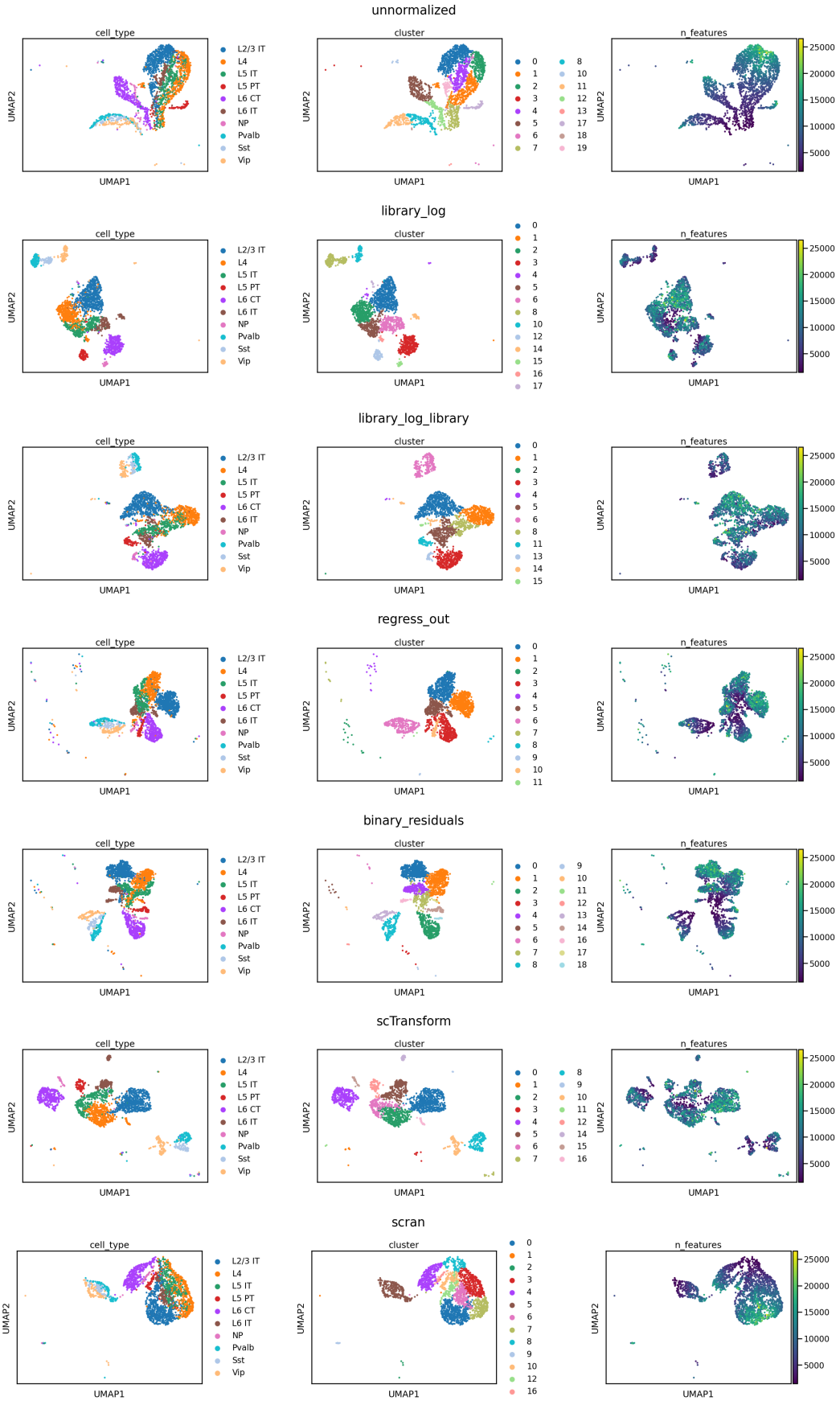

Figure 17. Umaps coloured by cell types, found leiden clusters and library size. Only annotated cells are depicted.

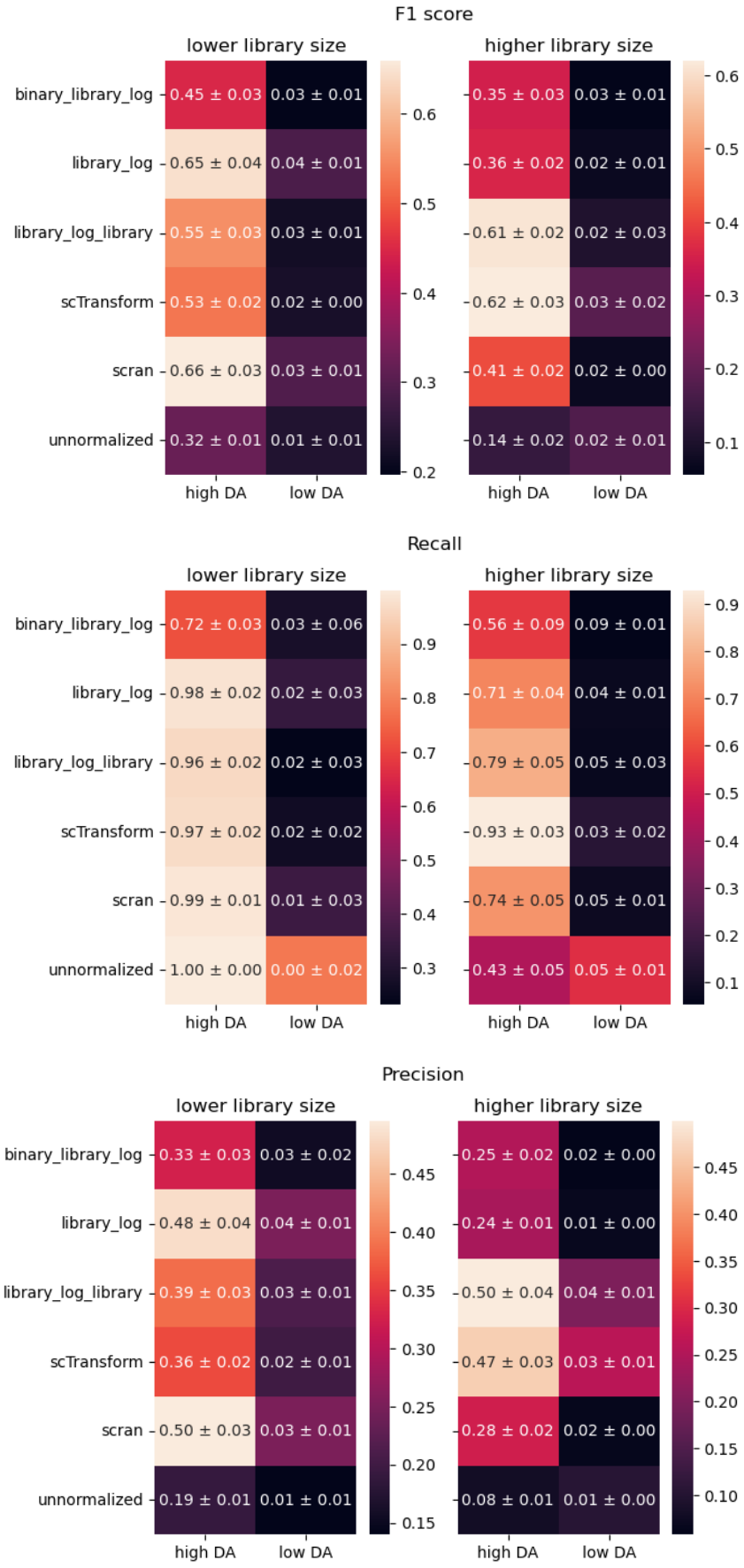

Figure 18. Heatmaps showing the performance of each normalization method for different DA configurations.