Evaluation strategies for scRNA-seq data integration

Hilal Kazan, PhD

Antalya Bilim University

23th of March, 2023

Workshop on scRNA-seq data integration, CA20117

Reference

nature methods

ANALYSIS

https://doi.org/10.1038/s41592-021-01336-8

OPEN

Benchmarking atlas-level data integration in single-cell genomics

Malte D. Luecken[1], M. Büttner[1], K. Chaichoompu[1], A. Danese[1], M. Interlandi[2], M. F. Mueller[1], D. C. Strobl[1], L. Zappia[1,3], M. Dugas[4], M. Colomé-Tatché[1,5,6] ✉ and Fabian J. Theis[1,3,5] ✉

Aim:

Methods

Datasets and preprocessing. We benchmarked data integration methods on 13 integration tasks: 11 real data tasks and two simulation tasks. For the real data tasks, we downloaded 23 published datasets (see Supplementary Data 2 for a per-batch overview of datasets). All scRNA-seq datasets were quality controlled and normalized in the same way according to published best practices[9]. Specifically, we used scran pooling normalization[40] (v.1.10.2 unless otherwise specified) and $\log_{+1}$ transformation on count data. For data solely available in transcripts per million or reads per kilobase of transcript, per million mapped reads units, we performed $\log_{+1}$ transformation without any further normalization. As the datasets typically contained different cell identity annotations we mapped these annotations by matching annotation names, overlaps of data-driven marker gene sets and manual clustering and annotation of cell identities per batch.

For the simulation tasks, data were simulated using the Splatter package[41] to evaluate data integration methods in a controlled setting. All of our data processing scripts are publicly available as Jupyter notebooks and R scripts at github.com/theislab/scib-reproducibility. For further details on datasets, please see the Supplementary Information.

Integration methods. We ran the 16 selected data integration methods according to default parameterizations obtained from available tutorials, paper methods or by directly contacting method authors. For further details on how each method was run, please see the Supplementary Information.

Metrics. We grouped the metrics into two broad categories: (1) removal of batch effects and (2) conservation of biological variance. The latter category is further divided into conservation of variance from cell identity labels, and conservation of variance beyond cell identity labels. Scores from the first category include principal component regression (batch), ASW (batch), graph connectivity, graph iLISI and kBET. In the second category, label conservation metrics include NMI, ARI, ASW (cell-type), graph cLISI, isolated label F1 and isolated label silhouette; label-free conservation metrics include cell-cycle (CC) conservation, HVG conservation and trajectory conservation.

The metrics were run on different output types (Supplementary Table 2). For example, metrics that run on *k*NN graphs can be run on all output types after preprocessing. Similarly, metrics that run on joint embeddings can also be run on corrected feature outputs. Preprocessing was performed in Scanpy (v.1.4.5 commit d69832a). *k*NN graphs were computed using the neighbors function where $k=15$ unless otherwise specified. Where a joint embedding was available, this graph was computed using Euclidean distances on this embedding, whereas distances were computed on the top 50 principal components where a corrected feature matrix was output.

NMI. NMI compares the overlap of two clusterings. We used NMI to compare the cell-type labels with Louvain clusters computed on the integrated dataset. The overlap was scaled using the mean of the entropy terms for cell-type and cluster labels. Thus, NMI scores of 0 or 1 correspond to uncorrelated clustering or a perfect match, respectively. We performed optimized Louvain clustering for this metric to obtain the best match between clusters and labels. Louvain clustering was performed at a resolution range of 0.1 to 2 in steps of 0.1, and the clustering output with the highest NMI with the label set was used. We used the scikit-learn[27] (v.0.22.1) implementation of NMI.

ARI. The Rand index compares the overlap of two clusterings; it considers both correct clustering overlaps while also counting correct disagreements between two clusterings[42]. Similar to NMI, we compared the cell-type labels with the NMI-optimized Louvain clustering computed on the integrated dataset. The adjustment of the Rand index corrects for randomly correct labels. An ARI of 0 or 1 corresponds to random labeling or a perfect match, respectively. We also used the scikit-learn[27] (v.0.22.1) implementation of the ARI.

ASW. The silhouette width measures the relationship between the within-cluster distances of a cell and the between-cluster distances of that cell to the closest cluster[43]. Averaging over all silhouette widths of a set of cells yields the ASW, which ranges between −1 and 1. The ASW is commonly used to determine the separation of clusters where 1 represents dense and well-separated clusters, while 0 or −1 corresponds to overlapping clusters (caused by equal between- and within-cluster variability) or strong misclassification (caused by stronger within-cluster than between-cluster variability), respectively.

To evaluate data integration outputs, we used (1) the classical definition of ASW to determine the silhouette of the cell labels (cell-type ASW) and (2) a modified approach to measure batch mixing. Both metrics were computed on the embeddings provided by integration methods or the PCA of expression matrices in case of feature output. For the bio-conservation score (1), the ASW was computed on cell identity labels and scaled to a value between 0 and 1 using the equation:

$$\text{cell type ASW} = (\text{ASW}_C + 1)/2,$$

where C denotes the set of all cell identity labels.

For the batch mixing score (2), we consider the absolute silhouette width, $s(i)$, on batch labels per cell i. Here, 0 indicates that batches are well mixed, and any deviation from 0 indicates a batch effect:

$$s_{\text{batch}}(i) = |s(i)|.$$

To ensure higher scores indicate better batch mixing, these scores are scaled by subtracting them from 1. As we expect batches to integrate within cell identity clusters, we compute the batchASW$_j$ (ref. [13]) score for each cell label j separately, using the equation:

$$\text{batch ASW}_j = \frac{1}{|C_j|}\sum_{i\in C_j} 1 - s_{\text{batch}}(i),$$

where C_j is the set of cells with the cell label j and $|C_j|$ denotes the number of cells in that set.

To obtain the final batchASW score, the label-specific batchASW$_j$ scores are averaged:

$$\text{batch ASW} = \frac{1}{|M|}\sum_{j\in M} \text{batch ASW}_j.$$

Here, M is the set of unique cell labels.

Overall, a batchASW of 1 represents ideal batch mixing and a value of 0 indicates strongly separated batches. We used the scikit-learn[27] (v.0.22.1) implementation to compute these scores.

Principal component regression. Principal component regression, derived from PCA, has previously been used to quantify batch removal[13]. Briefly, the R^2 was calculated from a linear regression of the covariate of interest (for example, the batch variable B) onto each principal component. The variance contribution of the batch effect per principal component was then calculated as the product of the variance explained by the ith principal component (PC) and the corresponding $R^2(\text{PC}_i|B)$. The sum across all variance contributions by the batch effects in all principal components gives the total variance explained by the batch variable as follows:

$$\text{Var}(C|B) = \sum_{i=1}^{G} \text{Var}(C|\text{PC}_i) \times R^2(\text{PC}_i|B),$$

where $\text{Var}(C|\text{PC}_i)$ is the variance of the data matrix C explained by the ith principal component.

Graph connectivity. The graph connectivity metric assesses whether the *k*NN graph representation, G, of the integrated data directly connects all cells with the same cell identity label. For each cell identity label c, we created the subset *k*NN graph $G(N_c,E_c)$ to contain only cells from a given label. Using these subset *k*NN graphs, we computed the graph connectivity (GC) score using the equation:

$$\text{GC} = \frac{1}{|C|}\sum_{c\in C}\frac{|\text{LCC}(G(N_c,E_c))|}{|N_c|}.$$

Here, C represents the set of cell identity labels, $|\text{LCC}()|$ is the number of nodes in the largest connected component of the graph and $|N_c|$ is the number of nodes with cell identity c. The resultant score has a range of $(0;1]$, where 1 indicates that all cells with the same cell identity are connected in the integrated *k*NN graph and the lowest possible score indicates a graph where no cell is connected. As this score is computed on the *k*NN graph, it can be used to evaluate all integration outputs.

kBET. The kBET algorithm (v.0.99.6, release 4c9dafa) determines whether the label composition of a k nearest neighborhood of a cell is similar to the expected (global) label composition[11]. The test is repeated for a random subset of cells, and the results are summarized as a rejection rate over all tested neighborhoods. Thus, kBET works on a *k*NN graph.

We computed *k*NN graphs where $k=50$ for joint embeddings and corrected feature outputs via the Scanpy preprocessing steps (previously described). To test for technical effects and to account for cell-type frequency shifts across datasets, we applied kBET separately on the batch variable for each cell identity label. Using the kBET defaults, a k equal to the median of the number of cells per batch within each label was used for this computation. Additionally, we set the minimum and maximum thresholds of k to 10 and 100, respectively. As *k*NN graphs that have been subset by cell identity labels may no longer be connected, we computed kBET per connected component. If >25% of cells were assigned to connected components too small for kBET computation (smaller than $k\times3$), we assigned a kBET score of 1 to denote poor batch removal. Subsequently, kBET scores for each label were averaged and subtracted from 1 to give a final kBET score.

We noted that k-nearest-neighborhood sizes can differ between graph-based integration methods (for example, Conos and BBKNN) and methods in which the *k*NN graph is computed on an integrated embedding. This difference can affect the test outcome because of differences in statistical power across neighborhoods.

Thus, we implemented a diffusion-based correction to obtain the same number of nearest neighbors for each cell irrespective of integration output type (Supplementary Note 1). This extension of kBET allowed us to compare integration results on *k*NN graphs irrespective of integration output format.

Graph LISI. The LISI, a diversity score, was proposed to assess both batch mixing (iLISI) and cell-type separation (cLISI)[21]. LISI scores are computed from neighborhood lists per node from integrated *k*NN graphs. Specifically, the inverse Simpson's index is used to determine the number of cells that can be drawn from a neighbor list before one batch is observed twice. Thus, LISI scores range from 1 to N, where N is the total number of batches in the dataset.

Typically, neighborhood lists to compute LISI scores are extracted from weighted *k*NN graphs with $k=90$ nearest neighbors at a fixed perplexity of $p=\frac{1}{3}k$. These nearest neighbor graphs are constructed using Euclidean distances on PCA or other embeddings. In contrast, integrated graphs that are output by methods such as Conos or BBKNN typically contain far fewer than $k=90$ neighbors. Running LISI metrics with differing numbers of nearest neighbors per node results in differing sensitivities per neighborhood and thus skews any comparison with graph-based integration outputs. Thus, the original LISI score is not applicable to graph-based outputs.

To extend LISI graph-based integration outputs, we developed graph LISI, which uses the integrated graph structure as an embedded space for distance calculation. The calculated graph distances are then used to determine a consistent number of nearest neighbors per node. We used the shortest path lengths computed via a custom scalable reimplementation of Dijkstra's algorithm[44] as a graph-based distance metric (see Supplementary Note 2 for details). Our graph LISI extension produces consistent metric values with the standard LISI implementation for non-graph-based integration outputs (Supplementary Fig. 41). Additionally, we sped up graph LISI scoring via a fast, parallel C++ implementation that scales to millions of cells.

As LISI scores range from 1 to B (where B denotes the number of batches), indicating perfect separation and perfect mixing, respectively, we rescaled them to the range 0 to 1. For iLISI and cLISI this involved a two-step process. First, we computed the median across neighborhoods per method: $\text{cLISI} = \text{median}\, f(x), x \in X$; $\text{iLISI} = \text{median}\, g(x), x \in X$. Second, we rescaled the LISI scores as follows: $\text{cLISI} : f(x) = \frac{B-x}{B-1}$, where a 0 value corresponds to low cell-type separation and $\text{iLISI} : g(x) = \frac{x-1}{B-1}$, where a 0 value corresponds to low batch integration.

Isolated label scores. We developed two isolated label scores to evaluate how well the data integration methods dealt with cell identity labels shared by few batches. Specifically, we identified isolated cell labels as the labels present in the least number of batches in the integration task. The score evaluates how well these isolated labels separate from other cell identities.

We implemented the isolated label metric in two versions: (1) the best clustering of the isolated label (F1 score) and (2) the global ASW of the isolated label. For the cluster-based score, we first optimize the cluster assignment of the isolated label using the F1 score across louvain clustering resolutions ranging from 0.1 to 2 in resolution steps of 0.1. The optimal F1 score for the isolated label is then used as the metric score. The F1 score is a weighted mean of precision and recall given by the equation:

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

It returns a value between 0 and 1, where 1 shows that all of the isolated label cells and no others are captured in the cluster. For the isolated label ASW score, we compute the ASW of isolated versus nonisolated labels on the PCA embedding (ASW metric above) and scale this score to be between 0 and 1. The final score for each metric version consists of the mean isolated score of all isolated labels.

HVG conservation. The HVG conservation score is a proxy for the preservation of the biological signal. If the data integration method returned a corrected data matrix, we computed the number of HVGs before and after correction for each batch via Scanpy's highly_variable_genes function (using the 'cell ranger' flavor). If available, we computed 500 HVGs per batch. If fewer than 500 genes were present in the integrated object for a batch, the number of HVGs was set to half the total genes in that batch. The overlap coefficient is as follows:

$$\text{overlap}(X,Y) = \frac{|X \cap Y|}{\min(|X|,|Y|)},$$

where X and Y denote the fraction of preserved informative genes. The overall HVG score is the mean of the per-batch HVG overlap coefficients.

Cell-cycle conservation. The cell-cycle conservation score evaluates how well the cell-cycle effect can be captured before and after integration. We computed cell-cycle scores using Scanpy's score_cell_cycle function with a reference gene set from Tirosh et al.[45] for the respective cell-cycle phases. We used the same set of cell-cycle genes for mouse and human data (using capitalization to convert between the gene symbols). We then computed the variance contribution of the resulting S and $G2/M$ phase scores using principal component regression (Principal component regression), which was performed for each batch separately. The differences in variance before, $\text{Var}_{\text{before}}$, and after, $\text{Var}_{\text{after}}$, integration were aggregated into a final score between 0 and 1, using the equation:

$$\text{CC conservation} = 1 - \frac{|\text{Var}_{\text{after}} - \text{Var}_{\text{before}}|}{\text{Var}_{\text{before}}}.$$

In this equation, values close to 0 indicate lower conservation and 1 indicates complete conservation of the variance explained by cell cycle. In other words, the variance remains unchanged within each batch for complete conservation, while any deviation from the preintegration variance contribution reduces the score.

Trajectory conservation. The trajectory conservation score is a proxy for the conservation of the biological signal. We compared trajectories computed after integration for certain clusters that had been manually selected during the data preprocessing step. Trajectories were computed using diffusion pseudotime implemented in Scanpy (sc.tl.dpt). We assumed that trajectories found in the unintegrated data for each batch gave the most accurate biological signal. Therefore, the starting cell of the trajectory, after integration, was defined by selecting the most extremal cell from the cell-type cluster that contained the starting cells of the pre-integration diffusion pseudotime, which was based on the first three diffusion components (see the immune cell task description for more details). Only cells from the largest connected component of the neighborhood graph were considered.

We computed Spearman's rank correlation coefficient, s, between the pseudotime values before and after integration (using the function pd.series.corr() in the Pandas[46] package; v.1.1.1). The final score was scaled to a value between 0 and 1 using the equation

$$\text{trajectory conservation} = (s+1)/2.$$

Values of 1 or 0 correspond to the same order of cells on the trajectory before and after integration or the reverse order, respectively. In cases where the trajectory could not be computed, which occurs when *k*NN graphs of the integrated data contain many connected components, we set the value of the metric to 0.

Ranking and metric aggregation. Metrics were run on the integrated and unintegrated AnnData[47] objects. We selected the metrics for evaluating performance based on the type of output data (Supplementary Table 2). For example, metrics based on corrected embeddings (Silhouette scores, principal component regression and cell-cycle conservation) were not run where only a corrected graph was output.

The overall score, $S_{\text{overall},i}$, for each integration run i was calculated by taking the weighted mean of the batch removal score, $S_{\text{batch},i}$, and the bio-conservation score, $S_{\text{bio},i}$, following the equation:

$$S_{\text{overall},i} = 0.6 \times S_{\text{bio},i} + 0.4 \times S_{\text{batch},i}.$$

In turn, these partial scores were computed by averaging all metrics that contribute to each score via:

$$S_{\text{bio},i} = \frac{1}{|M_{\text{bio}}|}\sum_{m_j\in M_{\text{bio}}} f(m_j(X_i)), \text{ and}$$

$$S_{\text{batch},i} = \frac{1}{|M_{\text{batch}}|}\sum_{m_j\in M_{\text{batch}}} f(m_j(X_i)).$$

Here, X_i denotes the integration output for run i and M_{bio} and M_{batch} denote the set of metrics that contribute to the bio-conservation and batch removal scores, respectively. Specifically, M_{bio} contains the NMI cluster/label, ARI cluster/label, cell-type ASW, isolated label F1 and silhouette, graph cLISI, cell-cycle conservation, HVG conservation and trajectory conservation metrics, while M_{batch} contains the PCR batch, batchASW, graph iLISI, graph connectivity and kBet metrics. To ensure that each metric is equally weighted within a partial score and has the same discriminative power, we min–max scaled the output of every metric within a task using the function $f()$, which is given by:

$$f(Y) = \frac{Y - \min(Y)}{\max(Y) - \min(Y)}.$$

Notably, using z scores (previously used for trajectory benchmarking[11]) instead of min–max scaling gives similar overall rankings (Spearman's $R>0.96$ for all tasks; using scipy.stats.spearmanr from Scipy[48] v.1.4.1). Our metric aggregation scheme follows best practices for ranking methods in machine learning benchmarks by taking the mean of raw metric scores before ranking[22]. Using this approach, we were able to compute comparable overall performance scores even when different numbers of metrics were computed per run.

Overall method rankings across tasks (for example, Fig. 3b) were generated from the overall scores for each method in each task (without considering simulation tasks). We ranked the methods in each task and computed an average rank across tasks. Methods that could not be run for a particular task were

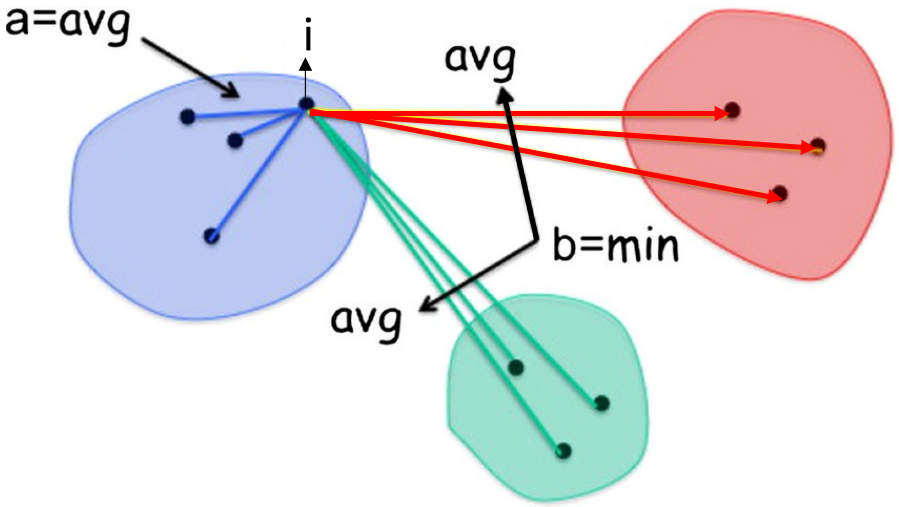
Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

Evaluation Metrics

- **Average Silhouette Width (Cell Type)**
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

Average Silhouette Width (ASW) (cell type)



Note: Clusters are determined based on cell identity labels

Silhouette width of a data point i

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

a(i): avg distance of i to other data points in the blue cluster

b(i): avg distance of i to other data points in the nearest cluster that doesn't contain i (green cluster)

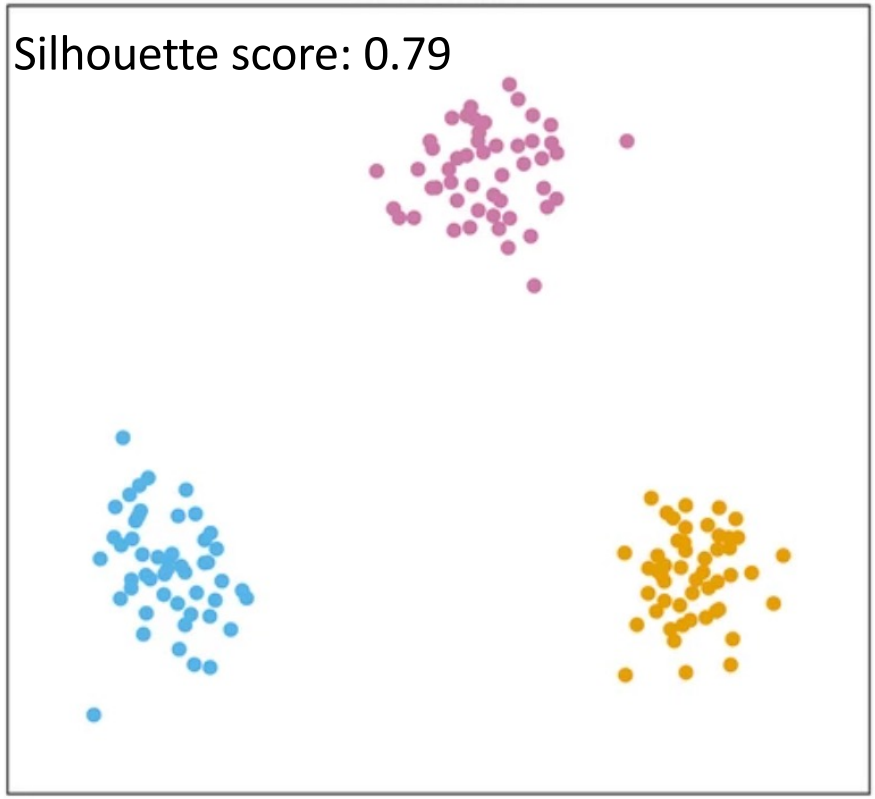
figure link: https://medium.com/@MrBam44/how-to-evaluate-the-performance-of-clustering-algorithms-3ba29cad8c03

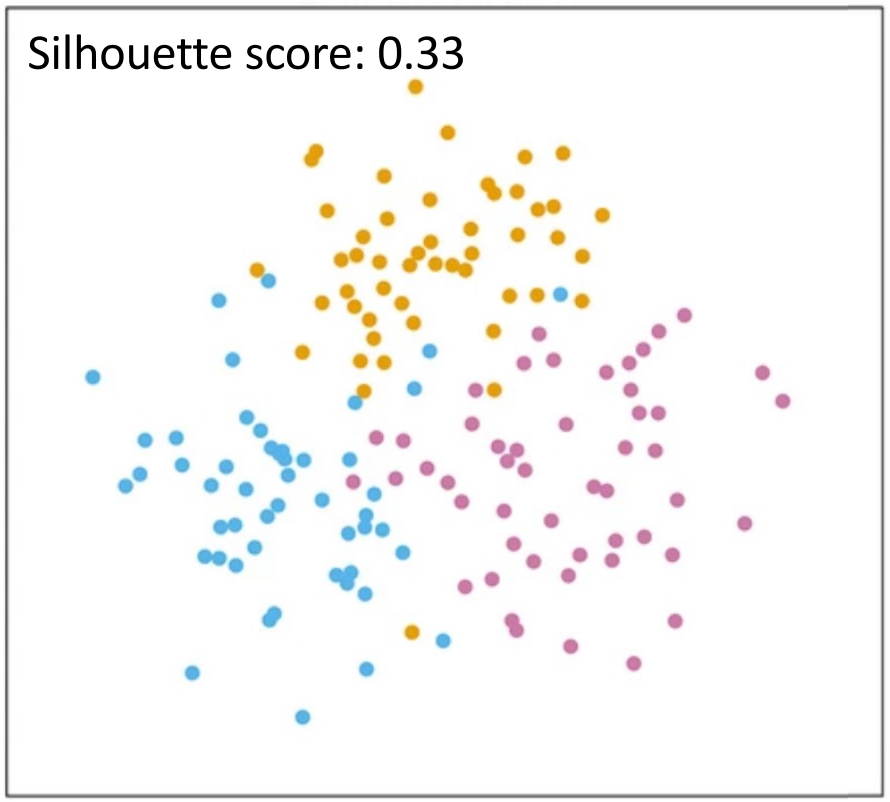
Average Silhouette Width (ASW) (cell type)

- **Silhouette width of the entire dataset:**

Average silhouette width across all the data points

Average Silhouette Width (ASW) (cell type)





figures taken from Dalmaijer et al, 2022

Average Silhouette Width (ASW) (cell type)

- **Silhouette width of the entire dataset:**

Average silhouette width across all the data points

- **Silhouette width is normally between -1 and 1.**

It's scaled between 0 (worst) and 1 (best) as follows:

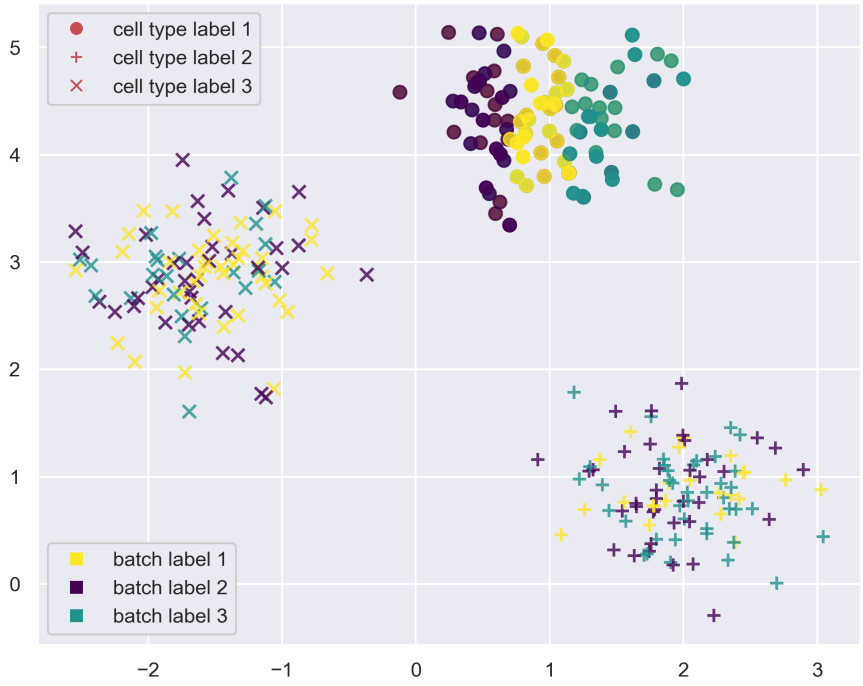
$$ASW_{scaled} = (ASW+1)/2$$

Evaluation Metrics

- Average Silhouette Width (Cell Type)
- **Average Silhouette Width (Batch)**
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

Average Silhouette Width (ASW) (batch)

Clusters are determined based on batch labels



1) $s_{\text{batch}}(i) = |s(i)|$ per cell i

deviations from 0 indicate batch effect

2) $\text{batch ASW}_j = \frac{1}{|C_j|} \sum_{i \in C_j} 1 - s_{\text{batch}}(i)$

C_j : set of cells with the cell type label j

ASW is computed for each cell type j separately.

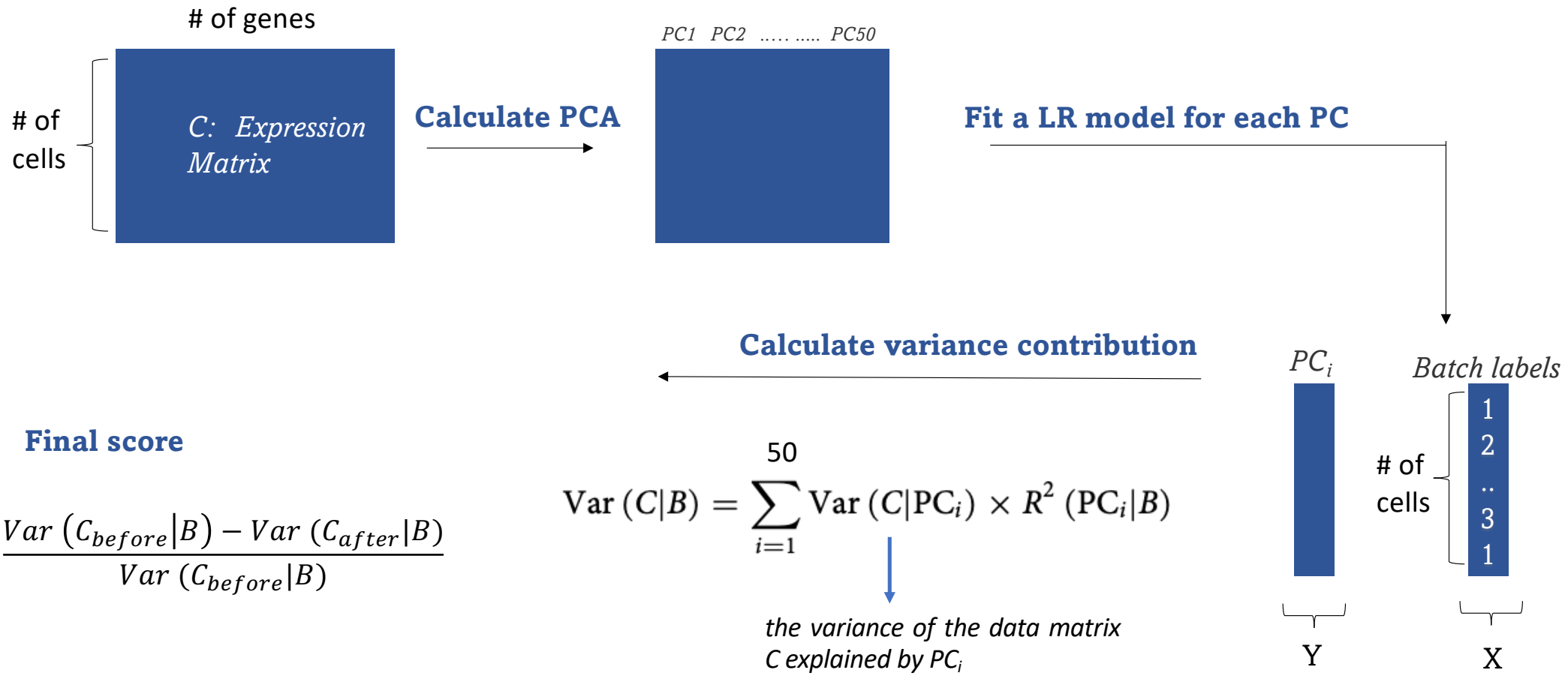
3) $\text{batch ASW} = \frac{1}{|M|} \sum_{j \in M} \text{batch ASW}_j$

M is the set of unique cell type labels

Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- **Principal Component Regression**
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

Principal component regression



Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- **kBET score**
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

kBET score

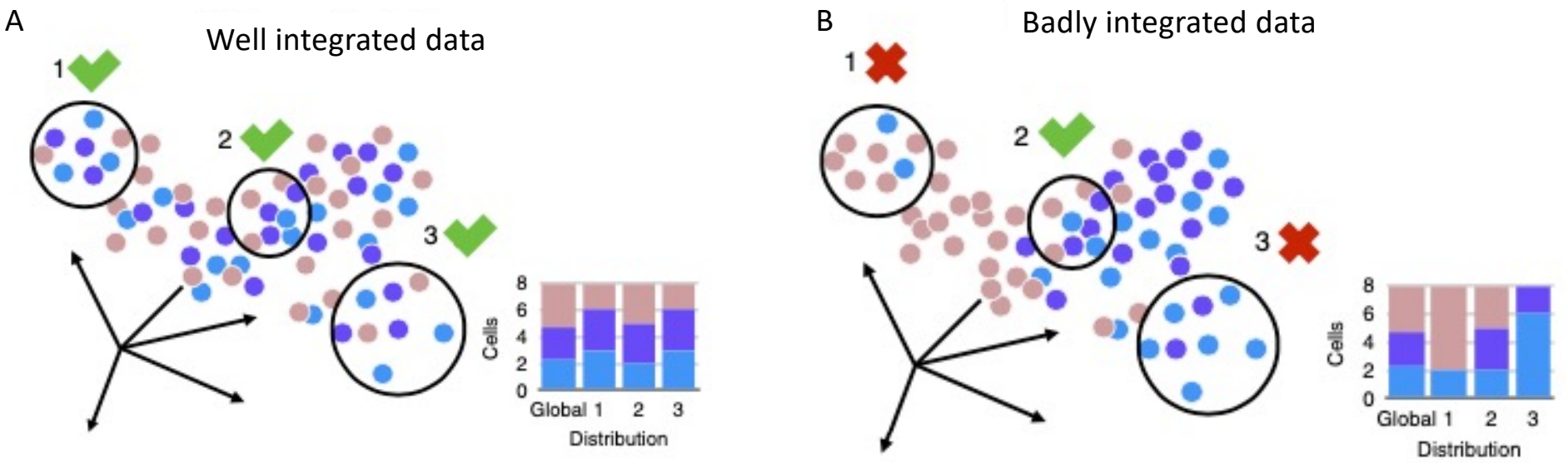
- kBET determines whether the batch label composition of a k-nearest neighborhood of a cell is similar to the expected (global) batch label composition

kBET score

- kBET determines whether the batch label composition of a k-nearest neighborhood of a cell is similar to the expected (global) batch label composition
- calculated separately for each cell type.
  - construct the kNN graph of a cell type
  - for randomly chosen 100 cells, perform chi-square test between local and global batch distribution. calculate avg rejection rate.

kBET score

- kBET determines whether the batch label composition of a k-nearest neighborhood of a cell is similar to the expected (global) batch label composition
- calculated separately for each cell type.
  - construct the kNN graph of a cell type
  - for randomly chosen 100 cells, perform chi-square test between local and global batch distribution. calculate avg rejection rate.



*colors represent batch labels.

Büttner et al, 2019

Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- **Graph Connectivity**
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

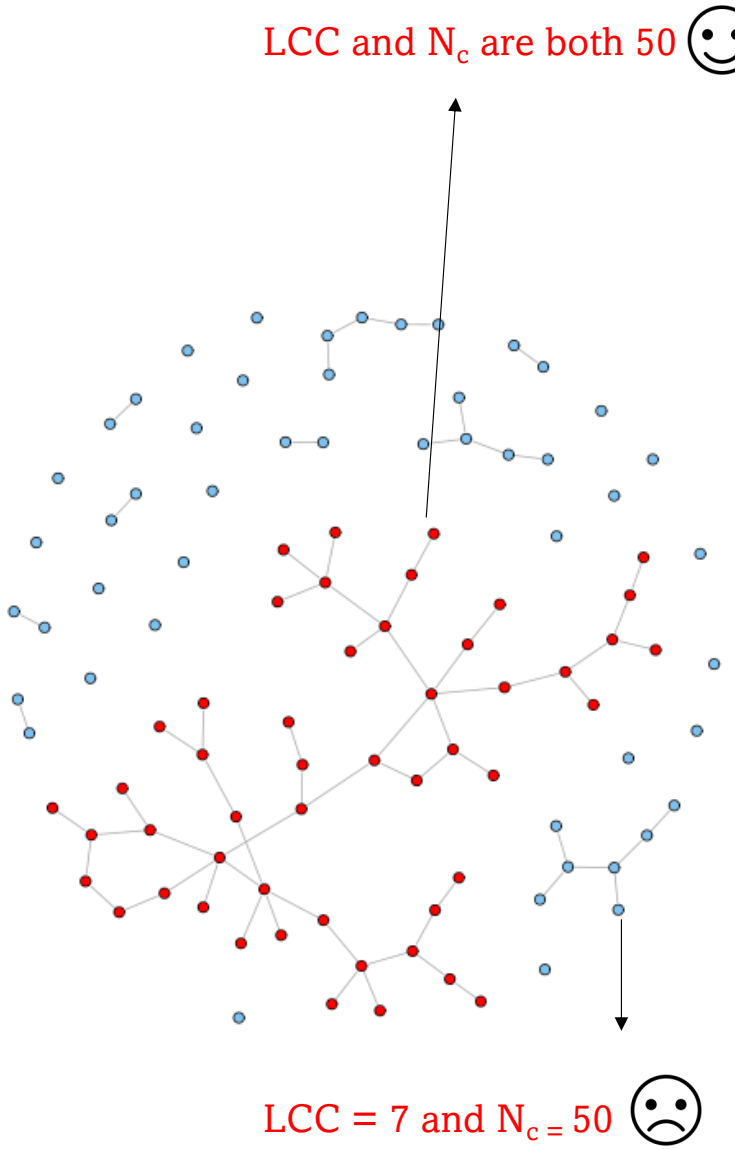
Graph connectivity

for each cell type label c:

The subset kNN graph $G(N_c;E_c)$ contains only cells from a given label c.

$$GC = \frac{1}{|C|} \sum_{c \in C} \frac{|LCC\,(G\,(N_c;E_c))|}{|N_c|}.$$

C: set of cell type labels
|LCC(G)|: the number of nodes in the largest connected component of the graph G
|Nc|: the number of nodes with cell type c.



Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- **Adjusted Rand Index**
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

Adjusted Rand Index (ARI)

Contingency table

$X \backslash Y$	Y_1	Y_2	$\cdots$	Y_s	sums
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sums	b_1	b_2	$\cdots$	b_s	

n_{ij} denotes the number of objects in common between X_i and Y_j.
Then ARI can be found by following formula:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}$$

ARI: 0.72

	Y_1	**Y_2**	**Y_3**
X_1	0	2	25
X_2	28	0	5
X_3	2	28	0

ARI: 0.45

	Y_1	**Y_2**	**Y_3**
X_1	10	10	0
X_2	0	20	10
X_3	20	0	0
X_4	0	0	20

Evaluation Metrics

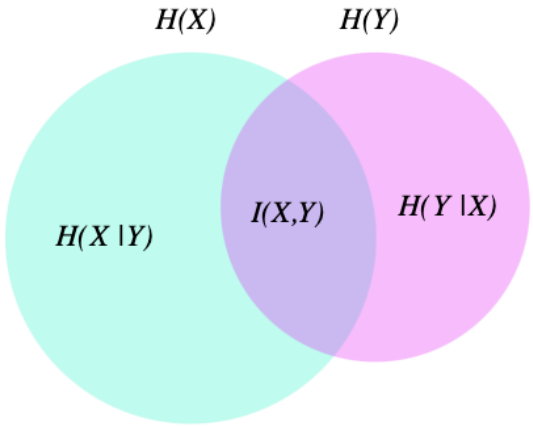
- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- **Normalized Mutual Information**
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

Normalized Mutual Information(NMI)

- MI quantifies the amount of information obtained about one random variable by observing the other random variable.
- For scRNA integration task, it measures the overlap between Louvain clustering and known cell types.

Normalized Mutual Information(NMI)

- MI quantifies the amount of information obtained about one random variable by observing the other random variable.
- For scRNA integration task, it measures the overlap between Louvain clustering and known cell types.



$$I(\mathrm{X};\mathrm{Y}) = H(\mathrm{X}) - H(\mathrm{X}|\mathrm{Y})$$

Normalized Mutual Information(NMI)

- MI quantifies the amount of information obtained about one random variable by observing the other random variable.
- For scRNA integration task, it measures the overlap between Louvain clustering and known cell types.

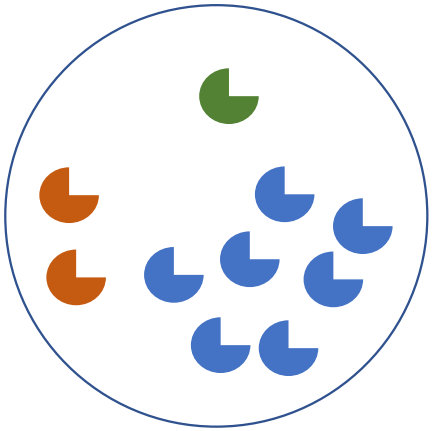
- Normalized Mutual Information:

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

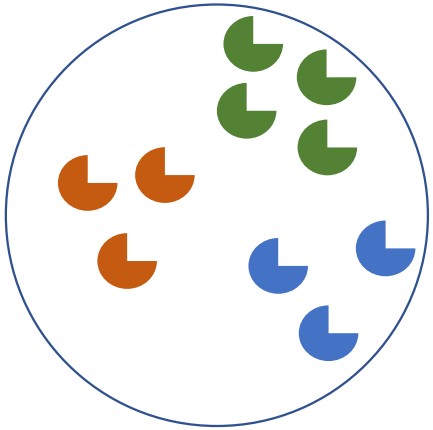
where, Y is the cell type labels and C is the Louvain cluster labels.
I(Y;C): Mutual Information between Y and C
H(Y): Entropy of cell types
H(C): Entropy of clusters

NMI scores of 0 representing no sharing/uncorrelated clustering and 1 representing perfect sharing of information/best match between Louvain clustering and cell labels.

NMI Example:



Cluster 2



Cluster 1

Cell type-1 (Y=1)
Cell type-2 (Y=2)
Cell type-3 (Y=3)

Entropy of the cell type:

P(Y=1)=5/20=1/4
P(Y=2) = 10/20 = 1/2
P(Y=3)=5/20=1/4
H(Y)= -1/4 • log(1/4)-1/4 • log(1/4)-1/2 • log(1/2)=1.5

Entropy of the clusters:

P(C=1) = 10/20 = 1/2
P(C=2) = 10/20 = 1/2
H(C) =-1/2 • log(1/2) -1/2 log • (1/2) = 1

Conditional Entropy:

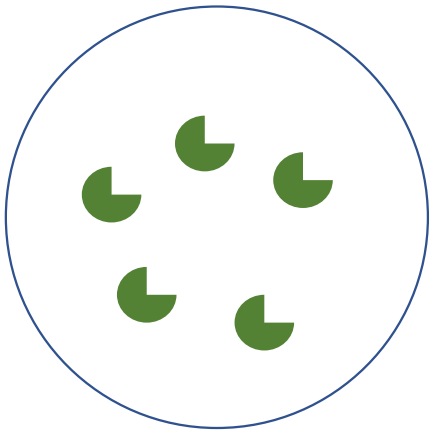
H(Y|C=1) = 0.78
H(Y|C=2) = 0.57

$$I(Y;C)=H(Y)-H(Y|C)$$

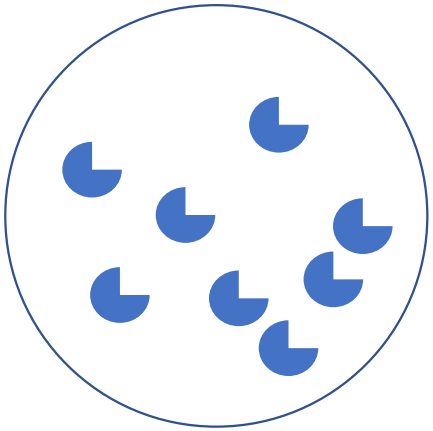
I(Y;C)= 1.5-(0.78+0.57)=0.13

NMI(Y;C)=(2×0.13)/(1.5+1)=0.108

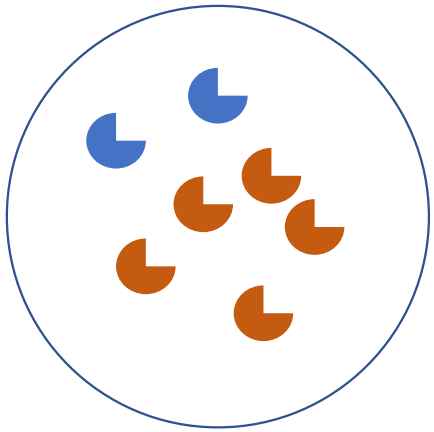
NMI Example:



Cluster 1



Cluster 2



Cluster 3

Cell type-1 (Y=1)

Cell type-2 (Y=2)

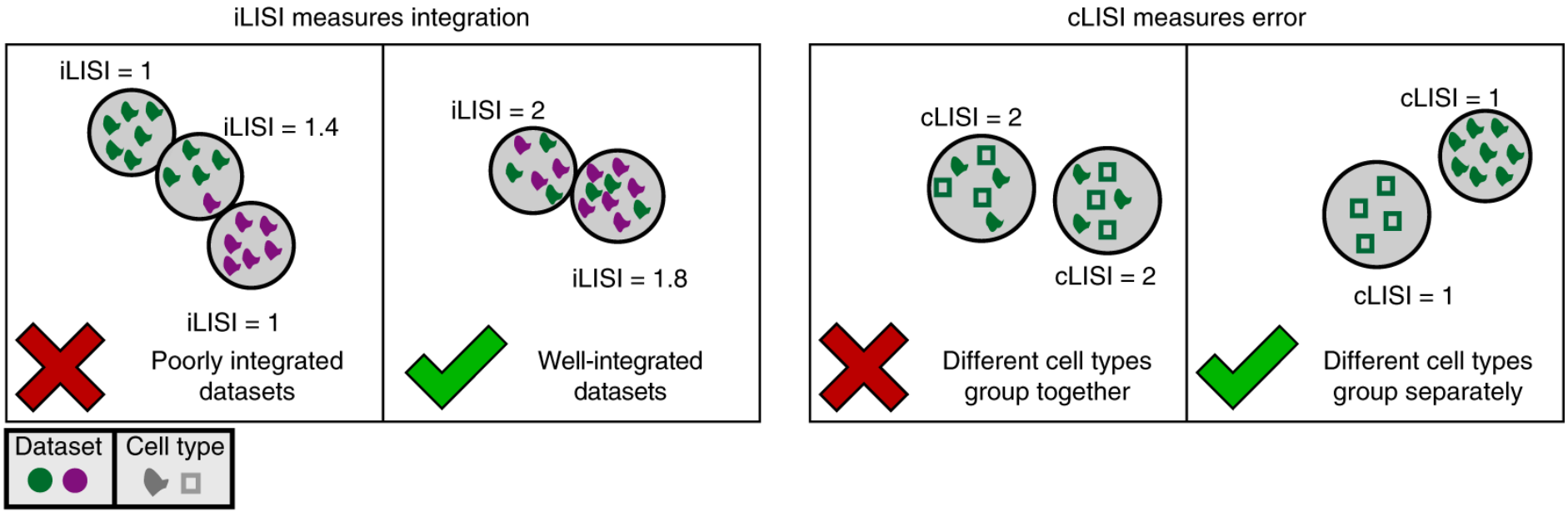
Cell type-3 (Y=3)

NMI: 0.78

Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- **iLISI and cLISI**
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

iLISI and cLISI



Korsunsky, I. et al. *Nature Methods 2019*

iLISI is scaled with g where g(x) = x-1 / B - 1

cLISI is scaled with f where f(x) = C-x / C-1

B: number of batches

C: number of cell types

Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- **Isolated Label Scores**
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

Isolated Label Scores

- Isolated cell labels are those labels that are present in the least number of batches.
- The score evaluates how well the isolated cell type is separated from the other cell types.

Cell Type	#of batches
....	...
alpha	6
beta	5
T cells	**4**

- T cells is selected as isolated cell label.
- If there are multiple isolated cell types, the scores are computed for each isolated type and then averaged over.

Isolated Label Scores

1) The best clustering of the isolated label (F1 score):

- Louvain clustering is performed with resolutions ranging from 0.1 to 2 in steps of 0.1.
- Among these clusterings, the one with the optimal F1 score is identified.

The F1 score is a weighted mean of precision and recall given by the equation:

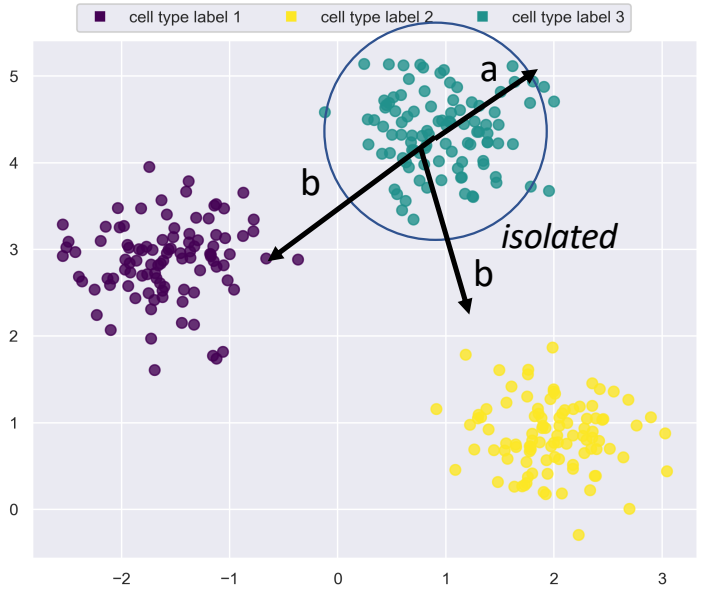
$$F1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

```
for each cluster label:
    y_pred = data["cluster"] == cluster_label
    y_true = obs["cell_type"] == "Tcell"
    f1 = f1_score(y_pred, y_true)
```

Isolated Label Scores

2) Isolated Label Silhouette Score

Calculate Average Silhouette Width score of isolated label over non-isolated labels.



$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- **Highly Variable Genes Conservation**
- Cell Cycle Conservation
- Trajectory Conservation

Highly Variable Genes (HVG) Conservation

- Metric computes the average percentage of overlapping highly variable genes per batch pre vs post integration.
- The score can only be computed on feature spaces.
- If available, 500 HVGs per batch computed.

The overlap coefficient is as

$$\text{overlap}(X, Y) = \frac{|X \cap Y|}{\min(|X|, |Y|)},$$

where X and Y denote the fraction of preserved informative genes.

- The overall HVG score is the mean of the per-batch HVG overlap coefficients.

Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- **Cell Cycle Conservation**
- Trajectory Conservation

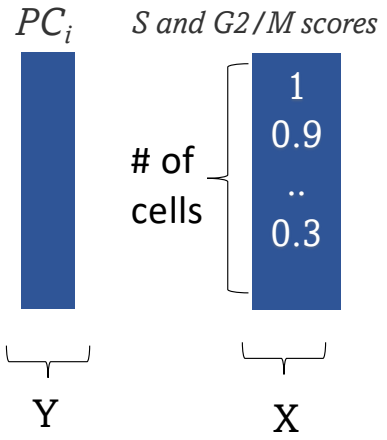
Cell Cycle Conservation

1 Compute the cell-cycle scores for the respective cell-cycle phases (G1, S or G2M) using a reference set of genes defined by Tirosh et al. Science 2016.

2 Then, compute the variance contribution of the resulting S and G2/M phase scores using principal component regression for each batch.

3 The differences in variance before and after integration were aggregated into a final score between 0 and 1 using the equation:

$$\text{CC conservation} = 1 - \frac{|\text{Var}_{\text{after}} - \text{Var}_{\text{before}}|}{\text{Var}_{\text{before}}}.$$



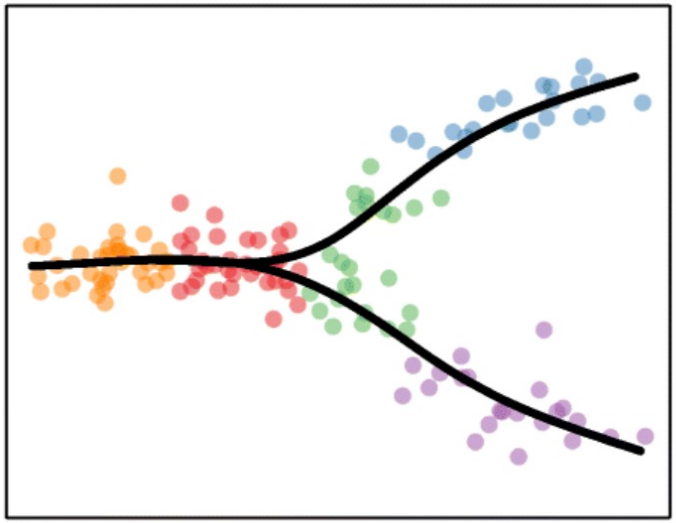
Evaluation Metrics

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- **Trajectory Conservation**

Trajectory conservation

- Spearman's rank correlation coefficient, *s*, is computed between the pseudotime values before and after integration. The final score was scaled to a value between 0 and 1 using the equation:

  trajectory conservation = (*s* + 1)/2.



Wikipedia

Ranking and metric aggregation

The overall score for each integration run i is calculated as a weighted mean:

$$S_{\text{overall},i} = 0.6 \times S_{\text{bio},i} + 0.4 \times S_{\text{batch},i}.$$

Partial scores are computed by averaging all metrics that contribute to each score via:

$$S_{\text{bio},i} = \frac{1}{|M_{\text{bio}}|} \sum_{m_j \in M_{\text{bio}}} f\left(m_j\left(X_i\right)\right), \text{ and}$$

$$S_{\text{batch},i} = \frac{1}{|M_{\text{batch}}|} \sum_{m_j \in M_{\text{batch}}} f\left(m_j\left(X_i\right)\right).$$

$$f(Y) = \frac{Y - \min(Y)}{\max(Y) - \min(Y)}.$$

M_{bio} and M_{batch} denote the set of metrics that contribute to the bio-conservation and batch removal scores.

Evaluation Metrics

M_{bio} M_{batch}

- Average Silhouette Width (Cell Type)
- Average Silhouette Width (Batch)
- Principal Component Regression
- kBET score
- Graph Connectivity
- Adjusted Rand Index
- Normalized Mutual Information
- iLISI and cLISI
- Isolated Label Scores
- Highly Variable Genes Conservation
- Cell Cycle Conservation
- Trajectory Conservation

+

Usability
Scalability

Acknowledgements

- S. Onur Doğan