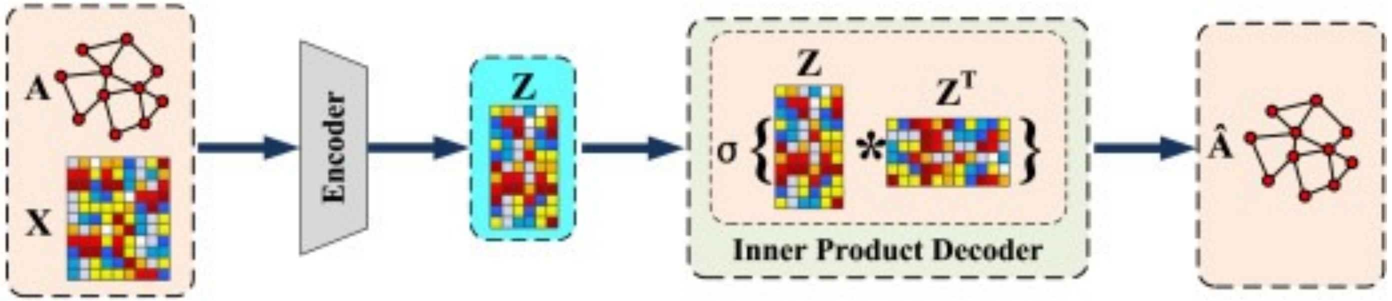
R-scGNN: Enhancing Graph Autoencoders for Improved Clustering in scRNA-seq Analysis

Shyaman Jayasundara

Graph Neural Networks for clustering

- GNNs for deconvoluting node relationships in a graph through neighbor information propagation

- Graph autoencoders learn a compact representation of the graph structure and capture node relationships from a global perspective, using a graph encoder/decoder architecture



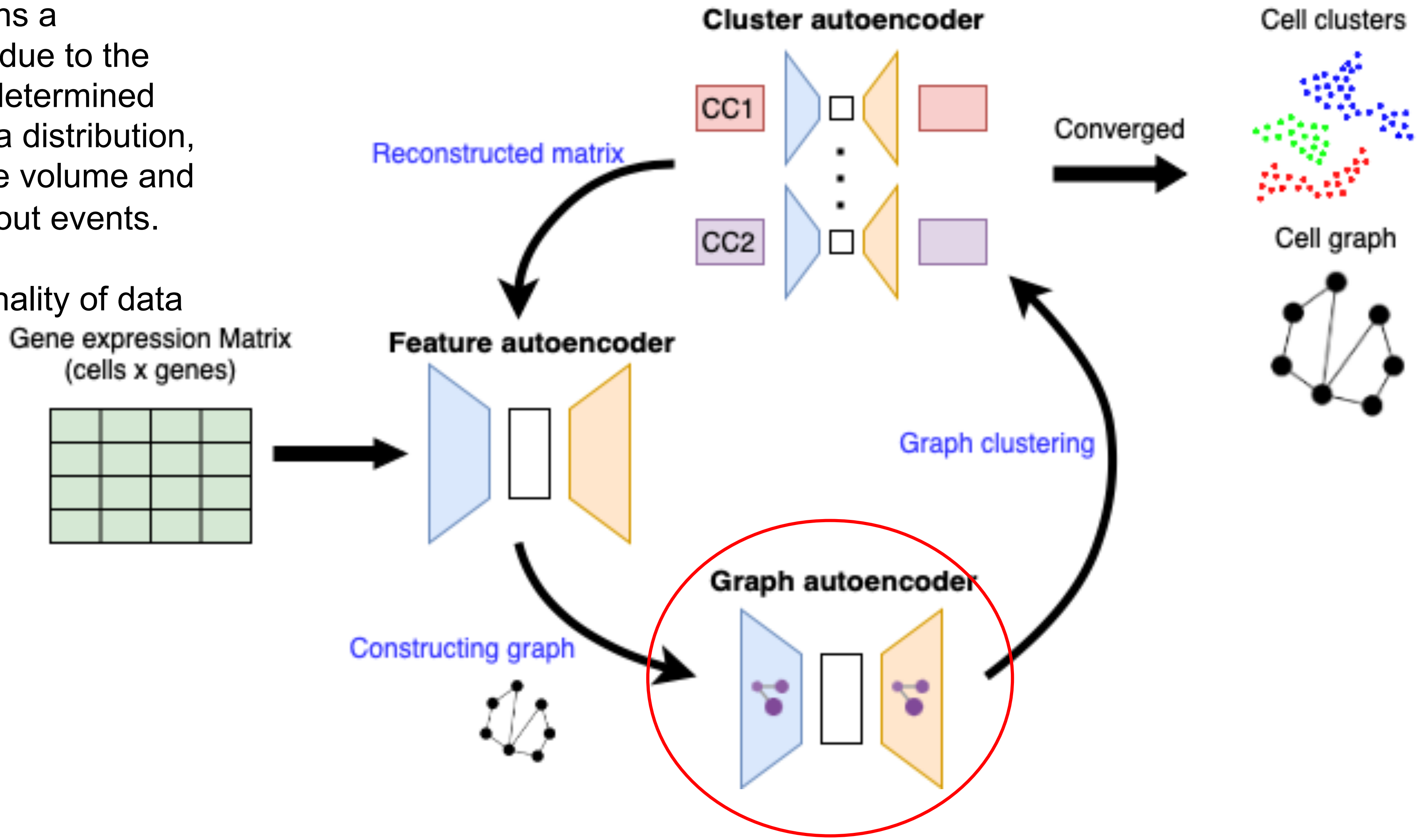
Sun et. al 2021

- The low-dimensional representation of nodes obtained from graph autoencoders can be used for clustering with various clustering algorithms.

scGNN: GNN framework for single-cell clustering

- Clustering remains a challenging task due to the complex and undetermined nature of the data distribution, which has a large volume and high rate of dropout events.

- Higher dimensionality of data



scGNN graph autoencoder

- scGNN uses a vanilla GAE (Kipf & Welling, 2016)

$$L_{\mathrm{GAE}} = L_{bce}(\hat{A}(Z(\theta)), A)$$

Z is graph embeddings and θ refers to the parameters of the model.

- scGNN separates clustering from the process of learning embedding

- scGNN has a limited capability to learn cluster-oriented features

Reformulate scGNN graph autoencoder

- Learn cluster-specific features by employing joint clustering and embedded learning

$$\theta^*, P^* = \arg\min_{\theta, P} L_{\text{clus}}\left(P(Z(\theta))\right),$$

P is the clustering assignments obtained by a certain clustering algorithm

$$\theta^*, P^* = \arg\min_{\theta, P} L_{\text{clus}}\left(P(Z(\theta))\right) + \gamma L_{bce}\left(\hat{A}(Z(\theta)), A\right)$$

- Two competing loss functions are optimized concurrently
  - clustering aims to decrease intra-cluster variance and increase inter-cluster variance
  - reconstruction objective which seeks to maintain all variances, including clustering-irrelevant similarities

Reformulate scGNN graph autoencoder

$$\theta^*, P^* = \arg\min_{\theta, P} L_{\text{clus}}\left(P(Z(\theta))\right) + \gamma L_{bce}\left(\hat{A}(Z(\theta)), A\right)$$

- Two competing loss functions are optimized concurrently
  - clustering aims to decrease intra-cluster variance and increase inter-cluster variance
  - reconstruction objective which seeks to maintain all variances, including clustering-irrelevant similarities
- This can arise an issue called **Feature Drift (FD)** (Mrabah et al., 2020)

- By optimizing θ, the embedded points are moved to create a clustering-oriented distribution. But embedded points may shift in a way that violates their semantic categories while still decreasing the embedded clustering penalty.
- Pseudo-supervision is needed to determine the semantic categories of the data by constructing pseudo-labels
- Training with pseudo-labels, a phenomenon known as **feature randomness (FR)** (Mrabah et al., 2020) can occur. Network may learn features that capture irrelevant similarities.

Reformulate scGNN graph autoencoder

$$\theta^*, P^* = \arg\min_{\theta,P} L_{\text{clus}}\left(P(Z(\theta))\right) + \gamma L_{bce}\left(\hat{A}(Z(\theta)), A\right)$$

- Tackle the FR and FD issues, Mrabah et al. (2022) proposed two solutions
  - sampling operator Ξ that gradually identifies nodes with reliable clustering assignments, to act as a protection mechanism against FR
  - graph-specific operator Υ that triggers a correction mechanism against FD

$$\theta^*, P^* = \arg\min_{\theta,P} L_{clus}(P(\Xi(Z(\theta)))) + \gamma L_{bce}(\hat{A}(Z(\theta)), \Upsilon(A, P(\Xi(Z(\theta))), \Omega))$$

R-scGNN

- scGNN framework's vanilla GAE was replaced with GMM-VGAE (Variational Graph Auto- Encoder with Gaussian Mixture Models) (Hui et al., 2020)

$$\begin{aligned} L_{\mathrm{R-GMM-VGAE}} = & L_{\mathrm{clus}}\left(P(\Xi(Z(\theta)))\right) \\ & + L_{bce}(\hat{A}(Z(\theta)), \Upsilon(A, P(\Xi(Z(\theta))), \Omega)) \end{aligned}$$

$$\begin{aligned} L_{clus}(P(Z(\theta))) = & \sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik} \log\left(\frac{\pi_k}{p_{ik}}\right) \\ & - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik}\left(\log\frac{\left|\operatorname{diag}\left(\sigma_k^2\right)\right|}{\left|\operatorname{diag}\left(\tilde{\sigma}_i^2\right)\right|}\right. \\ & + \operatorname{tr}\left(\operatorname{diag}^{-1}\left(\sigma_k^2\right)\operatorname{diag}\left(\tilde{\sigma}_i^2\right)\right) \\ & \left. + \left(\tilde{\mu}_i - \mu_k\right)^T \operatorname{diag}^{-1}\left(\sigma_k^2\right)\left(\tilde{\mu}_i - \mu_k\right) + d\right) \end{aligned}$$

Clustering performance metrics

- Adjusted Rand Index (**ARI**) - determine the similarities between all pairs of samples that were assigned to clusters in the current and previous clustering, adjusted by random permutation

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}$$

where the unadjusted Rand Index (RI) is $\frac{a+b}{C_2^n}$. a is the number of pairs correctly labeled in the same sets, b is the number of pairs correctly labeled as not in the same set, and C_2^n is the total number of possible pairs. $E[RI]$ is the expected RI of random labeling.

- **Silhouette** coefficient score - does not rely on known ground truth labels

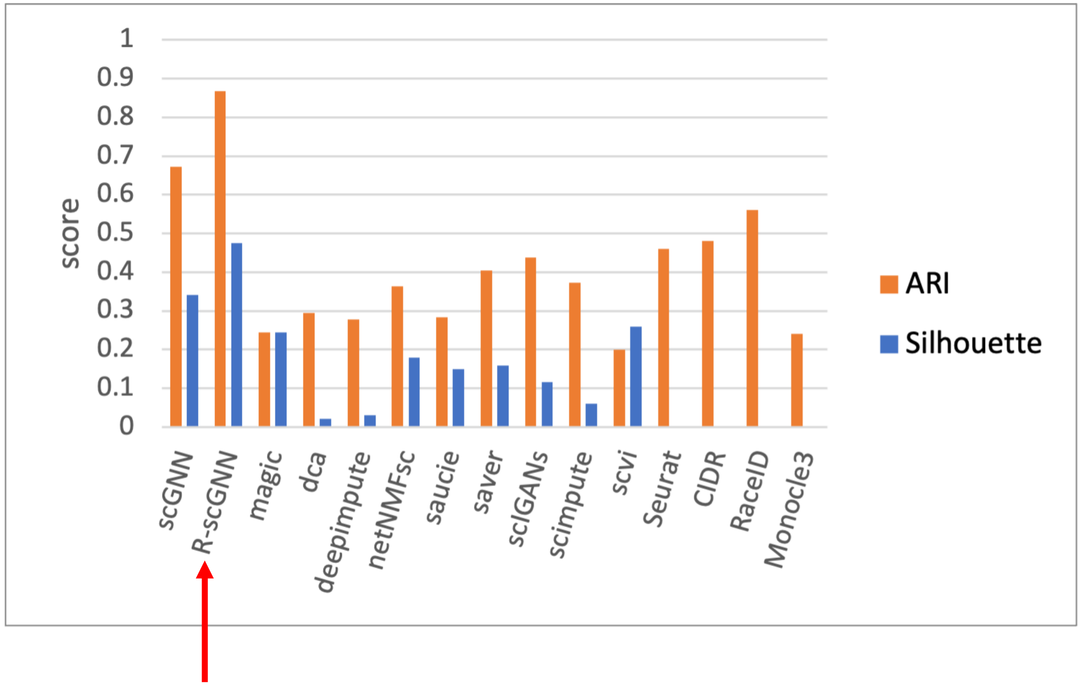
$$Silhouette = \frac{b - a}{\max(a, b)}$$

where a is the mean distance between a sample and all other points in the same cluster, and b is the mean distance between a sample and all other points in the next nearest cluster. The value of the Silhouette coefficient ranges from -1 to 1, where a score closer to 1 indicates better clustering.

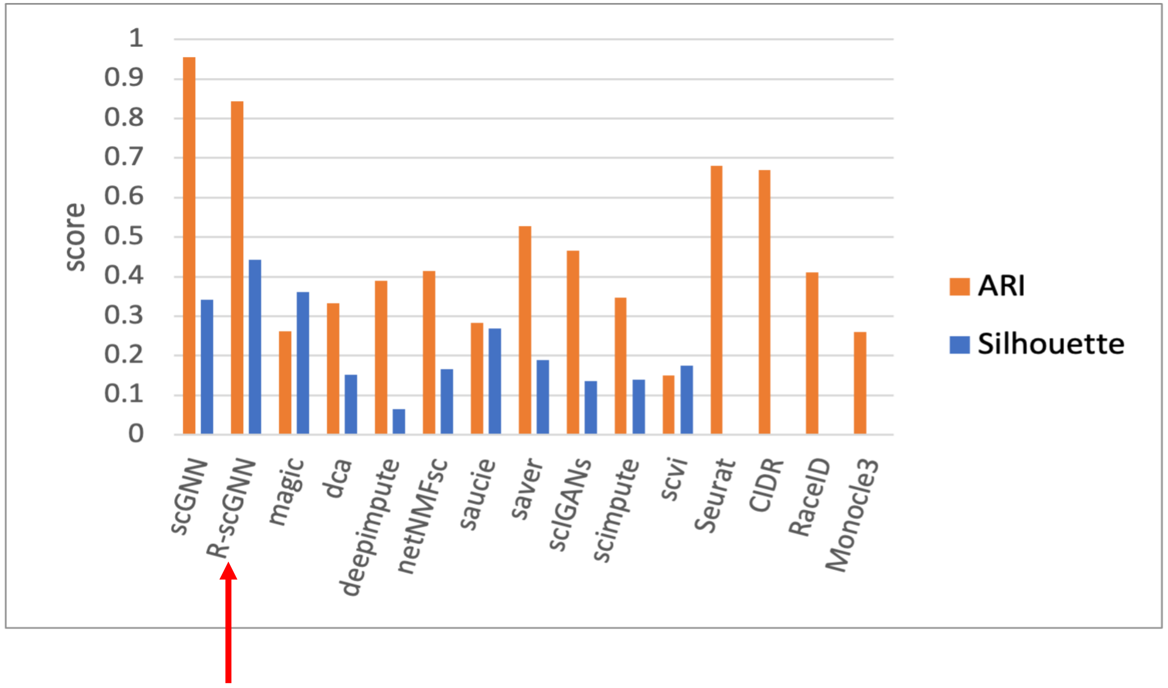
Results

- After running 10 iterations of the framework

Zeisel dataset

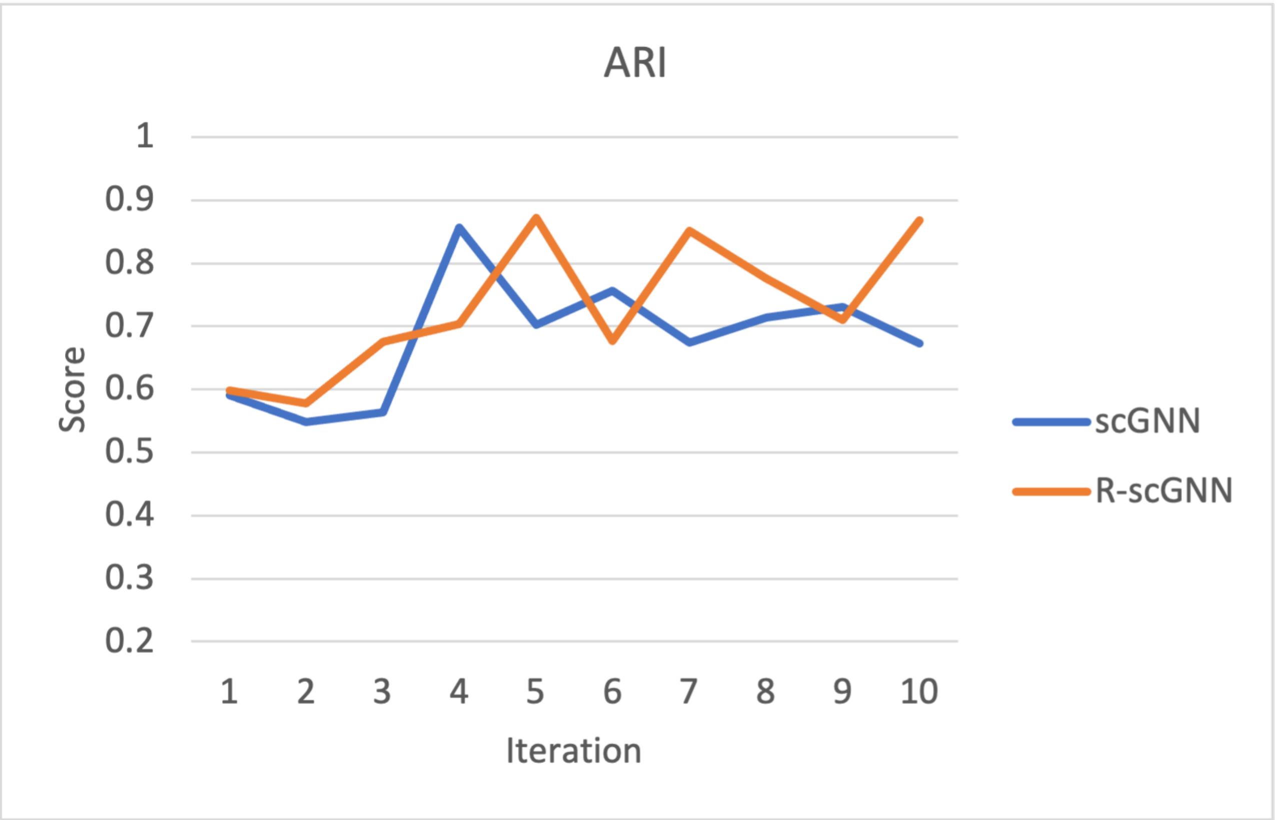


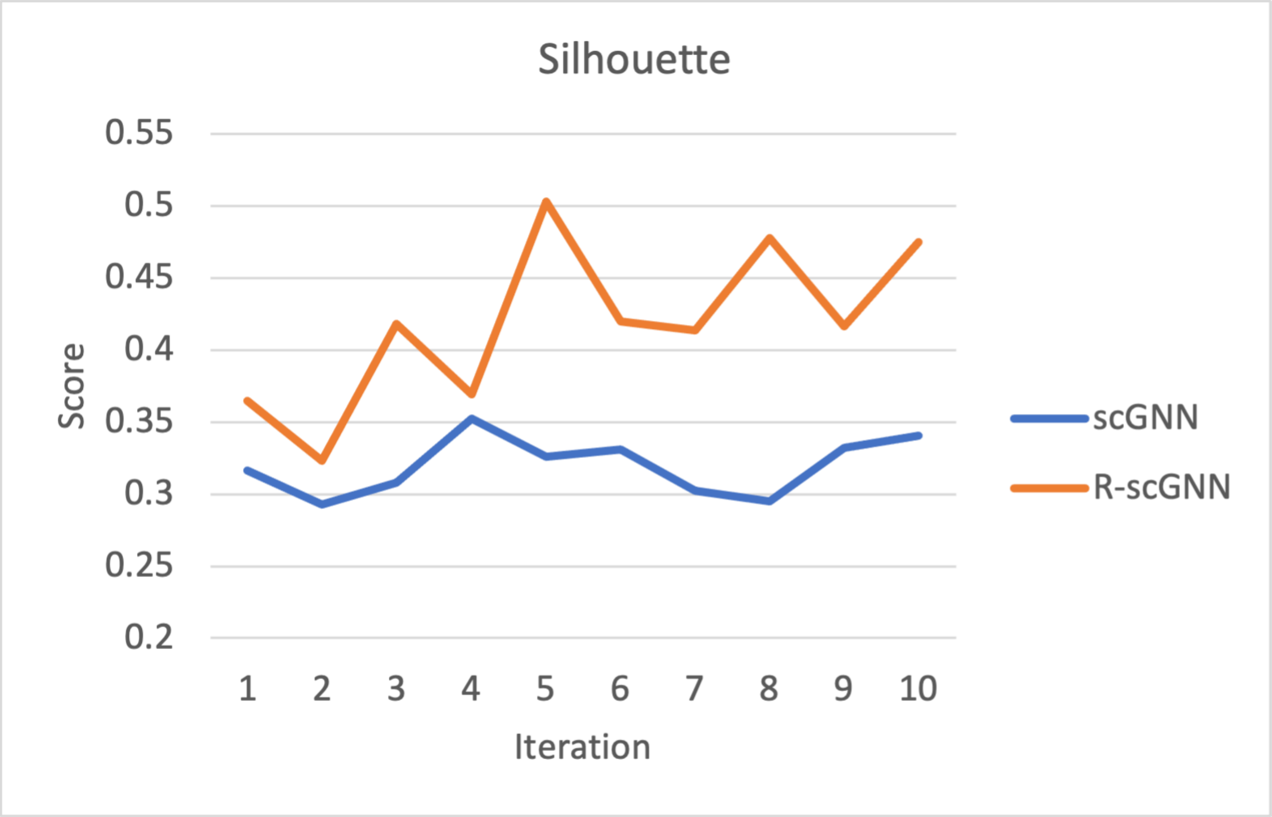
Klein dataset



Results

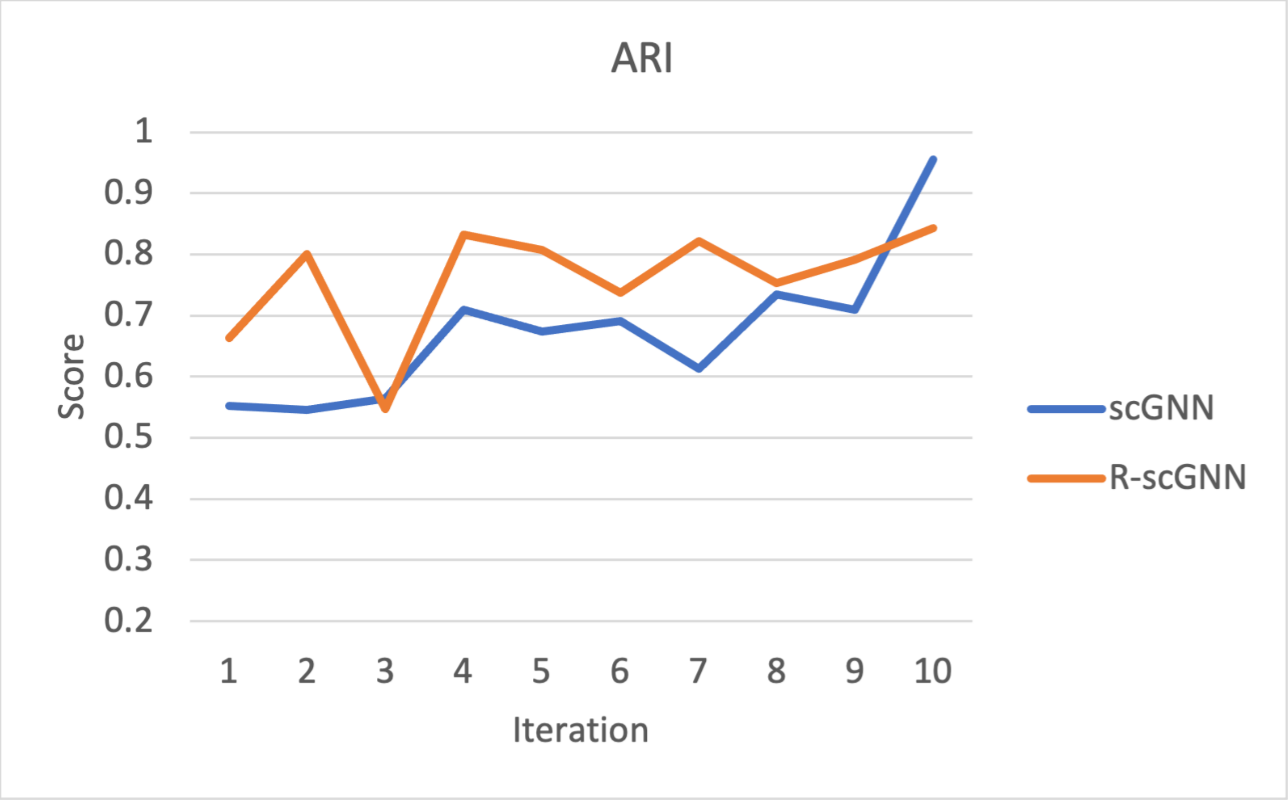
- Zeisel dataset

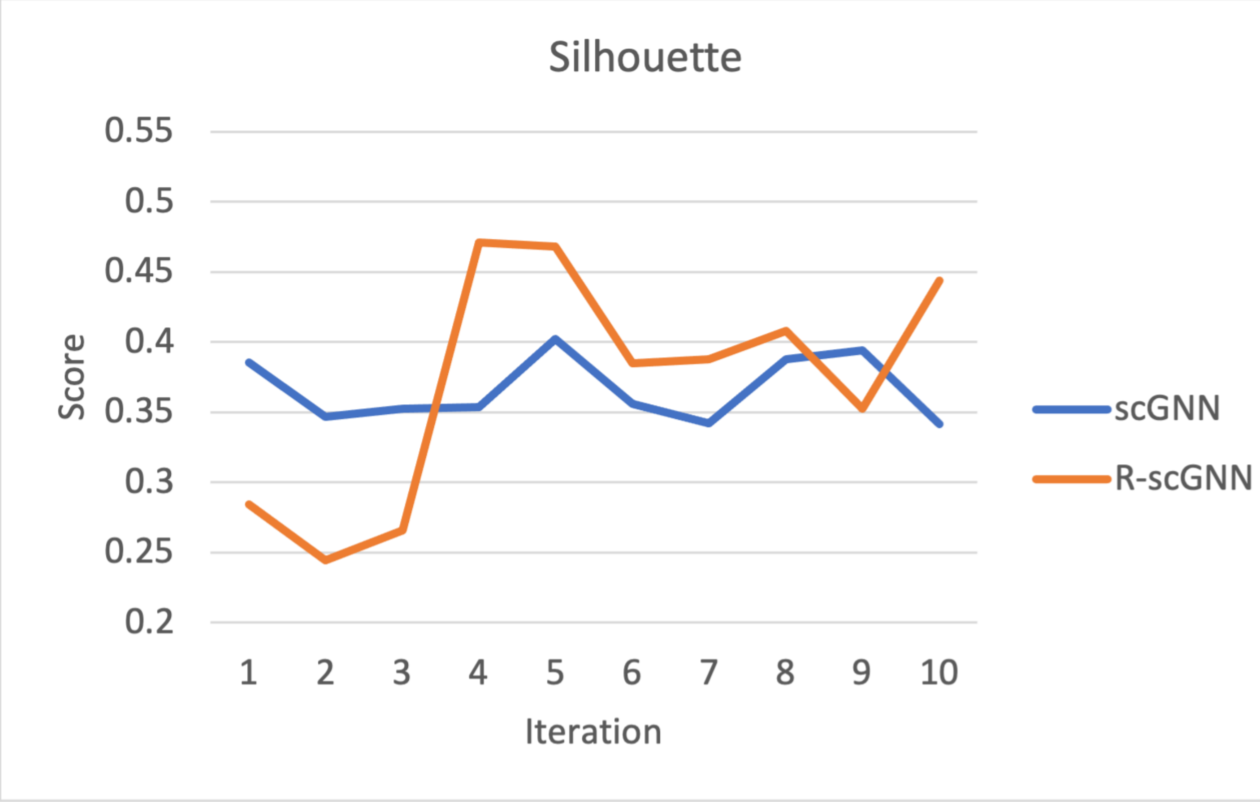




Results

- Klein dataset





Conclusion

- R-scGNN model outperforms other state-of-the-art methods in terms of clustering performance on scRNA-seq benchmark datasets

- Adjustment of GNN towards clustering objectives has resulted in improved performance

Thank you!