

# LSH-GAN: in-silico generation of cells for small sample high dimensional scRNA-seq data

Snehalika Lall

Indian Statistical Institute

Sumanta Ray (✉ [sumantababai86@gmail.com](mailto:sumantababai86@gmail.com))

University of Bielefeld <https://orcid.org/0000-0002-3371-8516>

Sanghamitra Bandyopadhyay

Indian Statistical institute

---

## Article

**Keywords:** scRNA-seq data, generative adversarial network, LSH-GAN

**Posted Date:** August 9th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-736403/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Version of Record:** A version of this preprint was published at Communications Biology on June 10th, 2022. See the published version at <https://doi.org/10.1038/s42003-022-03473-y>.

# LSH-GAN: in-silico generation of cells for small sample high dimensional scRNA-seq data

Snehalika Lall<sup>1</sup>, Sumanta Ray<sup>2,\*</sup>, and Sanghamitra Bandyopadhyay<sup>1,\*</sup>

<sup>1</sup>Machine Intelligence Unit, Indian Statistical Institute, Kolkata, West Bengal 700108, India.

<sup>2</sup>Genome Data Science, University of Bielefeld, Bielefeld, Germany.

\*Corresponding Author: sumanta.ray@cwil.nl, sanghami@isical.ac.in

## ABSTRACT

A fundamental problem of downstream analysis of scRNA-seq data is the unavailability of enough cell samples compare to the feature size. This is mostly due to the budgetary constraint of single cell experiments or simply because of the small number of available patient samples. Here, we present an improved version of generative adversarial network (GAN) called LSH-GAN to address this issue by producing new realistic cell samples. We update the training procedure of the generator of GAN using locality sensitive hashing which speeds up the sample generation, thus maintains the feasibility of applying the standard procedures of downstream analysis. LSH-GAN outperforms the benchmarks for realistic generation of quality cell samples. Experimental results show that generated samples of LSH-GAN improves the performance of the downstream analysis such as feature (gene) selection and cell clustering.

The corresponding software is available in: <https://github.com/Snehalikalall/LSH-GAN>

## Introduction

Recently, the emergence of high dimensional biological data such as single cell RNA sequence (scRNA-seq) data has posed a significant challenge to the machine learning researchers<sup>1,2</sup>. The high dimension, and small sample size (HDSS) data handling is difficult for downstream analysis particularly for feature selection (FS). It affects later stages of downstream analysis such as cell clustering, marker selection, and annotation of cell clusters. A few outliers can drastically affect the FS techniques, and the selected feature sets may not be adequate to discriminate the classes<sup>3</sup>. Moreover, high dimensionality increases the computational time beyond acceptability.

High dimensional small sample (HDSS) data is prevalent in the single cell domain due to the budgetary constraint, ethical consideration of single cell experiments, or simply because of the small number of available patient samples. Whatever the reason is, too few observations (cell sample) in the single cell data may create problems in the downstream analysis. This is because a small sample size may not reflect the whole population accurately, which surely degrades the performance of any model. The general pipeline of scRNA-seq downstream analysis starts with pre-processing (normalization, quality control) of the raw count matrix and then going through several steps which include identification of relevant genes, clustering of cells, and annotating cell clusters with marker genes<sup>4-8</sup>. Each step has a profound effect on the next stage of analysis. The gene selection step identifies the most relevant features/genes from the normalized/preprocessed data and has an immense impact on cell clustering. The general procedure for selecting relevant genes which are primarily based on high variation (highly variable genes)<sup>9,10</sup> or significantly high expression (highly expressed genes)<sup>4</sup> suffers from a small sample effect. The general FS techniques also failed to provide a stable and predictive feature set in this case due to an ultra large size of feature (gene). One way to solve this issue is to go for a robust and stable technique that does not overfit the data. A few attempts<sup>11-13</sup> were observed recently which embed statistical and information-theoretic approach. Although these methods result in stable features, however, these are not performed well in small sample scRNA-seq data.

Recently computational researchers gaining interest in this field. Some methods like cscGAN<sup>14</sup>, Splatter<sup>15</sup>, SUGAR<sup>16</sup> are already developed which uses different techniques (like generative model, statistical framework) to successfully simulate the samples of specific cell types or subpopulations. The challenge in this task is to handle the sparsity and heterogeneity of the cell populations which define the specific characteristics of scRNA-seq data. In this paper, we propose a generative model to sort out this problem in HDSS scRNA-seq data. We use generative adversarial model to generate realistic cell samples from a small number of available samples of HDSS scRNA-seq data. Generative Adversarial Network (GAN)<sup>17</sup> has already been shown to be a powerful technique for learning and generating complex distributions<sup>18,19</sup>. However, the training procedure of GAN is difficult and unstable. The training suffers from instability because both the generator and the discriminator model are trained simultaneously in a game that requires a Nash equilibrium to complete the procedure. Gradient descent does

this, but sometimes it doesn't, which results in a costly time consuming training procedure. The main contribution here is in modifying the generator input that results in a fast training procedure. We create a subsample of original data based on locality sensitive hashing (LSH) technique and augment this with noise distribution, which is given as input to the generator. Thus, the generator does not take pure noise as input, instead, we introduce a bias in it by augmenting a subsample of data with the noise distribution.

Researchers are still trying to find improved versions of the generative adversarial network (GAN) to use in different domains. Most of the variations such as progressive GAN (PGAN)<sup>20</sup>, Wasserstein GAN (WGAN)<sup>18</sup> try to train the model quicker than the conventional GAN. Unlike PGAN and WGAN, conditional GAN (CGAN)<sup>21</sup> operates by conditioning the conventional model on additional data sources (maybe class label or data from different modalities) to dictate the data generation. In our model, we direct our attention to the additional sample generation from HDSS data. However, the generated sample size becomes increasingly large with more features, the generation of which may not be feasible for conventional generative models. Augmenting subsample of real data distribution ( $p_{data}(x)$ ) with the prior noise ( $p_z(z)$ ) makes the training procedure of our model faster than the conventional GAN. We theoretically proved that the global minimum value of the virtual training criterion of the generator is less than the traditional GAN ( $< -\log 4$ ).

**Summary of contributions:** Here, we provide the following novelties:

- The proposed model is the first one to address the problem of downstream analysis (particularly gene selection and clustering) on HDSS scRNA-seq data.
- LSH-GAN can able to generate realistic samples in a faster way than the traditional GAN. This makes LSH-GAN more feasible to use in the feature (gene) selection problem of scRNA-seq data.
- LSH-GAN can produce more realistic cell samples than the other existing benchmarks.
- Here we derive a new way of training a generator that combines subsamples of original data with pure noise and takes this as input.
- For a fixed number of iteration LSH-GAN performed better than the traditional GAN in generating realistic samples.
- Gene selection and clustering on the generated samples of LSH-GAN provides excellent results for small-sample and large-feature sized single cell data.

## Results

In the following, we first describe the workflow of our analysis pipeline. Next, experimental validation of the proposed model is carried out by comparing it with several state-of-the-arts in real life scRNA-seq data. Finally, we used LSH-GAN to select genes from HDSS scRNA-seq data. We use benchmark gene selection techniques on the generated samples and used one single cell clustering technique to validate the selected genes.

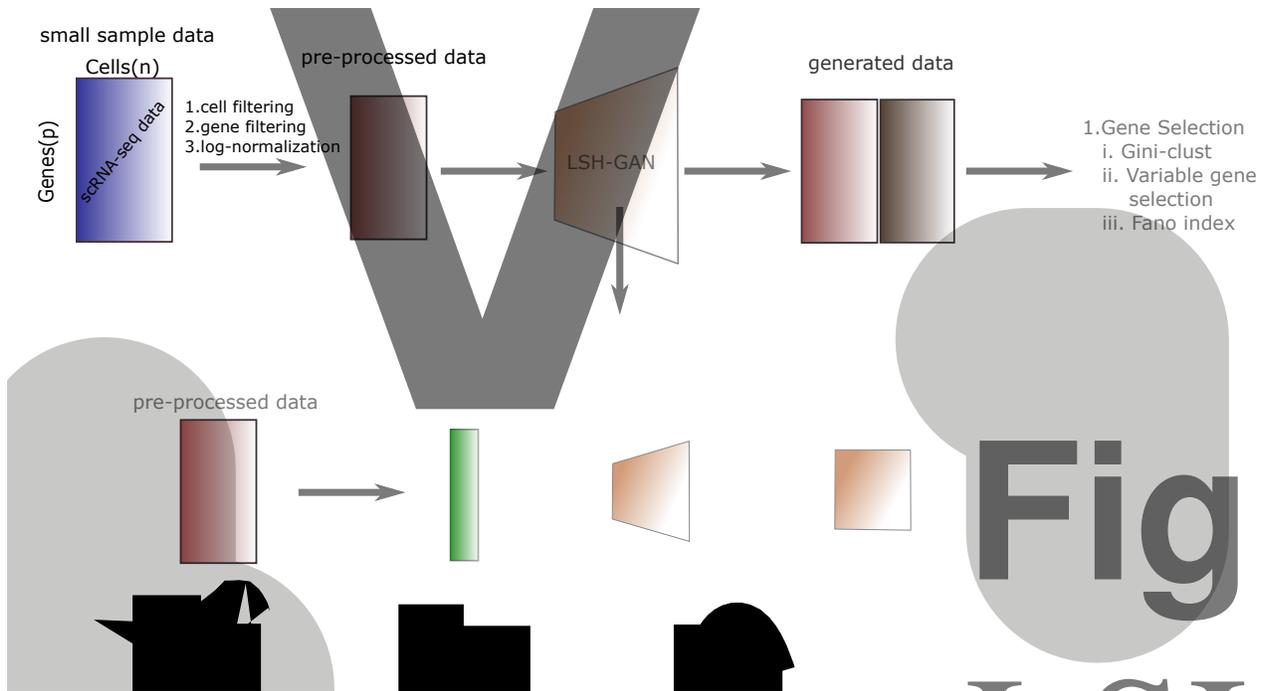
### Proposed model: LSH-GAN

The figure 1 describes the workflow of our analysis pipeline. Figure 1, panel-A, describes the application of the proposed LSH-GAN model in the feature selection task of the HDSS scRNA-seq data, while Panel-B depicts basic building blocks of the model. The following subsections describe in brief.

**LSH step: sampling of input data** Locality Sensitive Hashing (LSH)<sup>12,22,23</sup> is widely used in nearest neighbor searching to reduce the dimensionality of data. LSH utilizes locality-sensitive hash functions which hash similar objects into the same bucket with a high probability. The number of buckets is much lesser than the universe of possible items, thus reduces the search space of the query objects (see supplement for a detailed description of LSH technique).

In this work first, the unique hash codes which depict the local regions or neighborhoods of each data point are produced. For this, we utilized python sklearn implementation of *LSHForest* module with default parameters.

An approximate neighborhood graph ( $k$ -nn graph) is constructed by using  $k = 5$  for each data point. This step computes the euclidean distances between the query point and its candidate neighbors. Sampling is carried out in a 'greedy' fashion where each data point is *traversed* sequentially and its corresponding five nearest neighbors are flagged out which never visited again. Thus after one traversing a sub-set of samples is obtained which is further down-sampled by performing the same step iteratively.



---

**Algorithm** LSH-GAN Algorithm

---

**Input:** Data Matrix ( $\mathbf{x}$ ), number of Training iterations, number of nearest neighbor ( $\mathbf{k}$ ), number of iterations for sub-sampling ( $\mathbf{t}$ )

**Output:** Generated data ( $\mathbf{G}_{\text{out}}$ ).

- 1: **for** number of training iterations **do**
  - 2:  $x_s = \text{LSH-SAMPLING}(x, k, t)$
  - 3: augment  $p_{x_s}(x_s)$  with prior noise  $p_z(z)$  and give this ( $p_{\hat{z}}(\hat{z})$ ) to the generator,  $G$ .
  - 4: real data  $p_{\text{data}}(x)$  and generated data  $p_g(x)$  is given to discriminator  $D$ .  
**Update the Discriminator,  $D$**
  - 5:  $\Delta_d = \sum_{i=1}^n \log(D(x_i)) + \log(1 - D(G(\hat{z}_i)))$   
**Update the Generator,  $G$**
  - 6:  $\Delta_g = \sum_{i=1}^n \log(1 - D(G(\hat{z}_i)))$
  - 7: **end for**  
{The adaptive momentum gradient decent rule is used in our experiment.}
  - 8: **procedure** LSH-sampling( $x, k, t$ )
  - 9: Execute Locality Sensitive Hashing (LSH) on  $x$  and prepare a  $k$ -Nearest Neighbour list for each data point.
  - 10: **for** number of iteration of sub-sampling  $t$  **do**
  - 11: visit each data point sequentially in the order as it appears in data.
  - 12: if the data point is not visited earlier, select the data point and discard all its  $k$  neighbors from its nearest-neighbour list.
  - 13: **end for**
  - 14: **end procedure**
- 

Darmanis, Yan, Pollen, Klein, and Melanoma respectively (see supplementary figure-2). For generating hash code from LSH sampling, *LSHForest* of *scikit-learn* version 0.19.2 is utilized.

We take the adaptive learning rate optimization algorithm implemented in ADAM optimizer in python Tensorflow version 1.9.2. Generator ( $G$ ) and Discriminator ( $D$ ) uses 2-layer multilayer perceptrons with hidden layer dimension as (16, 16). For traditional GAN, we retain the same settings as LSH-GAN for  $G$  and  $D$  networks.

For benchmarking our method we have utilized three state-of-the-art techniques widely used for sample generation: cscGAN<sup>14</sup>, SUGAR<sup>16</sup>, and Splatter<sup>15</sup>. For these three methods, We adopted the code (with default parameters) provided on the Github page of the original publications.

Five well known gene selection methods (with default parameters) of scRNA-seq data are adopted for validation: *GLM-PCA*<sup>24</sup>, *CV<sup>2</sup> Index*, *M3Drop*<sup>25</sup>, *Fano Factor*<sup>26</sup> and Highly Variable Gene (HVG) selection of Seurat V4<sup>27</sup>. We select the top 500 features (genes) using all three feature selection methods on scRNA-seq datasets. For validation purposes, Wasserstein metric<sup>18</sup> is utilized to estimate the quality of the generated data. Clustering of scRNA-seq data is performed using *SC3*<sup>28</sup> technique with default parameters. Clustering performance is evaluated using the Adjusted Rand Index (ARI).

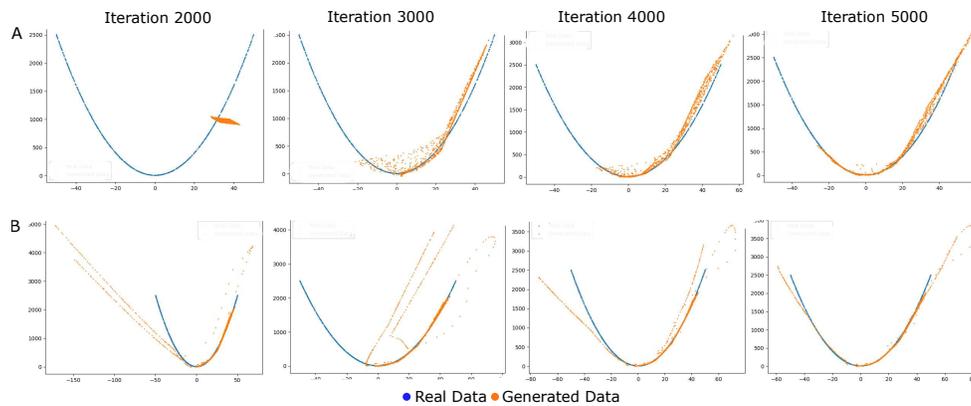
### LSH-GAN improves performance of traditional GAN on simulated data

First, we train the LSH-GAN on HDSS synthetic data and generate realistic samples to compare against the traditional GAN model. For this, we create a 2-class non-overlapping Gaussian mixture data consisting 100 samples and 1000 features by taking the mean ( $\mu$ ) of the data in the range of 5 to 15 for class-1 and  $-15$  to  $-5$  for class-2. The covariance matrix ( $\Sigma$ ) is taken for all the samples using the formula  $\Sigma = (\rho^{|i-j|})$ , where  $i, j$  are row and column index, and  $\rho$  is equal to 0.5. We calculate Wasserstein metric to estimate the quality of the generated data. The Wasserstein distance between the real data distribution ( $p_{\text{data}}$ ) and the generated data distribution ( $p_g$ ) to estimate the quality of the generated data. We use different settings of  $k^{\text{th}}$  ( $k=5, 10, 15, 20$ ) nearest neighbor to generate sub-sample of data from LSH sampling procedure. In each case, the sampled data ( $p_{x_s}$ ) is augmented with prior noise ( $p_z$ ) and given to the generator of LSH-GAN for model training.

For comparison with the traditional GAN model, we use the data with train: test split of 80:20 and calculate the Wasserstein metric between the test sample and the generated sample. Table 1 shows the values of the metric for LSH-GAN and traditional GAN model in different range of epochs and nearest neighbors  $k$ . A closer look into the table 1 reveals that the performance of LSH-GAN (at 10000 epoch and  $k=5$ ) is far better than the traditional GAN model with 25000 epochs. Notably, for less amount of sampling (larger  $k$ ), LSH-GAN needs more iterations for training. As for particular example, the performance of LSH-GAN achieved on 20000 epoch and  $k=20$  is rivaled only at 10000 epoch for  $k=10$ . Thus it is evident from the results that reducing the amount of sampling needs more epochs and thus needs more training time for the LSH-GAN model to converge. Figure 2 also supports this statement. Here, the two models (LSH-GAN, and traditional GAN) are trained to simulate a two dimensional synthetic data of known distribution, for which the LSH-GAN can able to generate samples that are more

**Table 1.** Wasserstein distance between generated and real data distribution. Model is trained on synthetic data of size  $100 \times 1000$  Gaussian mixture data with 2 non-overlapping classes.

Nearest Neighbour	Model	Epoch			
		10000	15000	20000	25000
$k = 5$	LSH-GAN	0.46	0.35	0.33	0.45
$k = 10$	LSH-GAN	1.09	0.89	0.83	0.82
$k = 15$	LSH-GAN	1.36	0.89	1.45	0.87
$k = 20$	LSH-GAN	1.53	1.35	1.19	0.83
	GAN	1.71	1.73	1.75	1.70



**Figure 2.** Generation of two dimensional synthetic data using traditional GAN (upper row, Panel-A) and LSH-GAN (lower row, Panel-B) model for different epochs.

real than the traditional GAN, in a lesser number of iteration.

### Comparison of LSH-GAN with benchmarks in HDSS scRNA-seq data

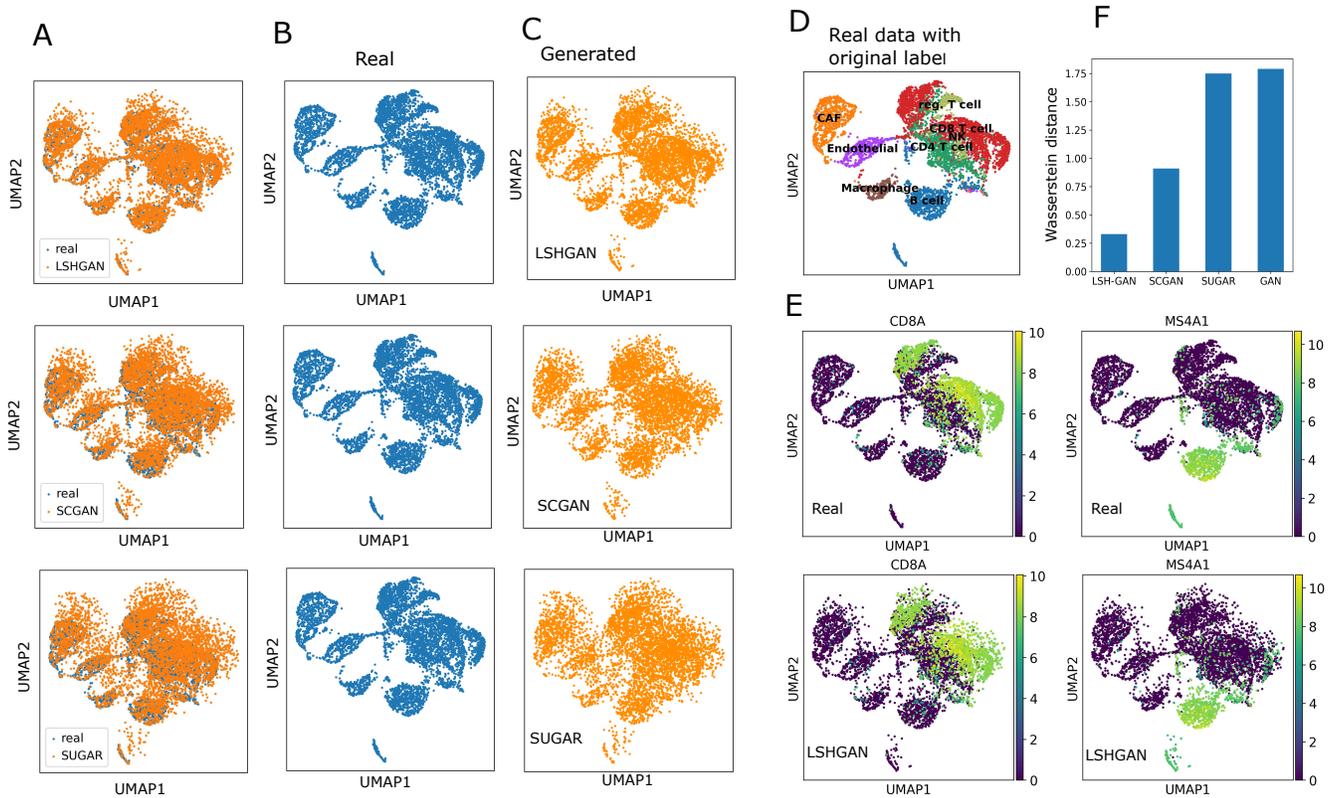
We compare LSH-GAN with four existing benchmarks: cscGAN<sup>14</sup>, splatter<sup>15</sup>, SUGAR<sup>16</sup> and traditional GAN<sup>17</sup>. Since the evaluation of the generative model is notoriously difficult<sup>29,30</sup>, we first use Wasserstein distance to compare the real data distribution and generated data distribution coming from different competing models. We also used UMAP visualization, and marker genes expression to visualize the generated cell samples. Figure 3, panel-A-C shows the two-dimensional UMAP representation of generated and real cell sample from the test data for four competing models. Melanoma data is utilized for this experiment. As can be seen from this figure, LSH-GAN can able to retain the distribution of the original cell samples. This can also be supported by the Wasserstein distance (see figure 3, panel-F) measured between real data and generated data distribution. To know how the expression of the marker genes are retained in the generated data, we plot the expression of marker gene CD8A (marker for CD8T cell) and MS4A1 (marker for B cell) in the two dimensional UMAP space for both the real and generated samples of LSH-GAN (see figure 3, panel-D-E). It reveals from the figure that marker genes CD8A and MS4A1 show similar expression patterns (high expression) both in real and generated cell samples.

We also validate the generated samples by training a classifier (random forest) to see whether it can able to distinguish the samples coming from two different distributions (real and generated). The aim is to see whether the model can discriminate between the real and generated cell samples accurately. Table 2 shows the cross validation AUC score of the random forest classifier for five scRNA-seq datasets. It reveals from the table that for LSH-GAN, the AUC scores hardly reach 0.6 (only for melanoma data) suggesting a chance-level performance of RF model. This suggests the generated data obtained from LSH-GAN is highly similar to the real data.

### Gene selection in HDSS scRNA-seq data

Here, we aim to address the problem of gene selection in HDSS scRNA-seq data using the generated samples. We augment the generated sample with original data to make the sample to feature ratio as 1.5. The augmented data is utilized for gene selection. Here, we have employed five feature selection methods (Highly Variable Gene (HVG) selection of Seurat V3/V4, M3Drop, GLM-PCA and *Fano Factor*,  $CV^2$ -index), widely used for the gene selection task in scRNA-seq data and one single cell clustering method (SC3) technique to validate the selected genes from the augmented data.

LSH-GAN is compared with five other state-of-the-arts in four HDSS scRNA-seq datasets (Darmanis, Yan, Pollen, and



**Figure 3.** Panel–A–C: UMAP visualization of real and generated cell samples of melanoma data. Panel-D shows real data with the original labels. Panel-E shows the expression of two markers CD8A (marker of CD8 T cell) and MS4A1 (marker of B cell) in real and generated data. Panel-F shows a barplot which describe the Wasserstein distance between the generated and real cell sample.

**Table 2.** Table shows results of applying random forest classifier for discriminating real and generated samples coming from different competing methods. The average AUC score (with 5-fold cross validation) is reported for each dataset.

	AUC Score				
	Yan	Darmanis	Pollen	Klein	Melanoma
cscGAN	0.65 ±0.02	0.68 ±0.01	0.64 ±0.02	0.62 ±0.02	0.66 ±0.01
Splatter	0.69 ±0.01	0.69 ±0.02	0.67 ±0.03	0.65 ±0.01	0.72 ±0.02
SUGAR	0.67 ±0.02	0.66 ±0.03	0.61 ±0.02	0.64 ±0.02	0.68 ±0.01
GAN	0.72 ±0.02	0.71 ±0.02	0.73 ±0.02	0.72 ±0.02	0.76 ±0.03
LSH-GAN	0.59 ±0.01	0.60 ±0.02	0.58 ±0.02	0.57 ±0.01	0.60 ±0.01

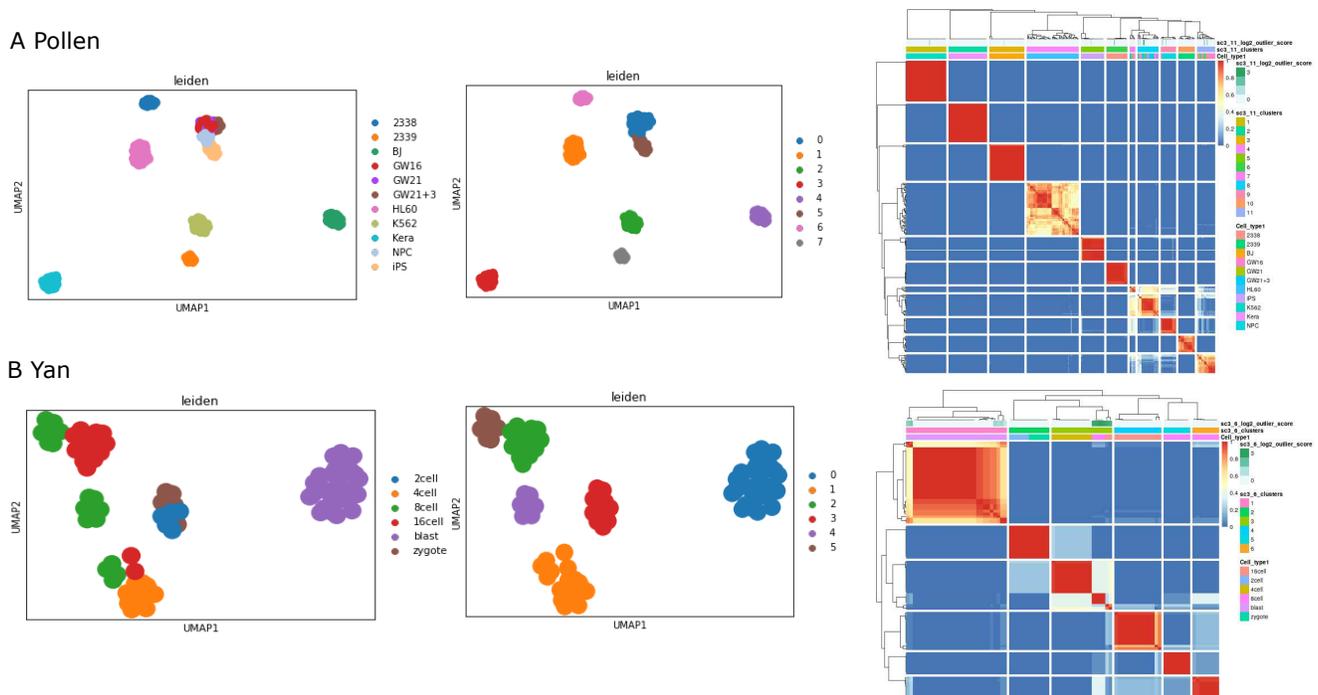
**Table 3.** Table shows the Adjusted Rand Index (ARI) scores of clustering results on the cell samples generated by the five competing methods.

Data	FS method	LSH-GAN	SUGAR	cscGAN	Splatter	GAN	without model
Darmanis	GLM-PCA	0.634	0.413	0.531	0.420	0.129	0.4
	Fano Factor	0.535	0.319	0.457	0.380	0.270	0.340
	CV <sup>2</sup> index	0.598	0.420	0.510	0.481	0.461	0.457
	M3Drop	0.648	0.513	0.580	0.507	0.480	0.460
	HVG (Seurat V4)	0.680	0.510	0.556	0.539	0.460	0.430
Yan	GLM-PCA	0.895	0.709	0.798	0.715	0.62	0.66
	Fano Factor	0.821	0.790	0.801	0.768	0.730	0.713
	CV <sup>2</sup> index	0.891	0.801	0.825	0.793	0.719	0.70
	M3Drop	0.898	0.802	0.796	0.790	0.761	0.71
	HVG (Seurat V4)	0.910	0.811	0.891	0.802	0.810	0.80
Pollen	GLM-PCA	0.835	0.780	0.819	0.793	0.788	0.780
	Fano Factor	0.933	0.878	0.916	0.880	0.815	0.712
	CV <sup>2</sup> index	0.94	0.906	0.908	0.890	0.831	0.81
	M3Drop	0.918	0.864	0.897	0.790	0.758	0.735
	HVG (Seurat V4)	0.958	0.916	0.897	0.868	0.801	0.820
Klein	GLM-PCA	0.815	0.769	0.784	0.731	0.581	0.66
	Fano Factor	0.8	0.742	0.782	0.770	0.669	0.796
	CV <sup>2</sup> index	0.82	0.710	0.761	0.709	0.690	0.680
	M3Drop	0.837	0.794	0.769	0.718	0.61	0.607
	HVG (Seurat V4)	0.898	0.861	0.857	0.785	0.730	0.739

Klein datasets). We exclude Melanoma data for this analysis as it already has larger sample size compared to the feature size (sample:feature is 3.46). The aim is to know whether the selected features/genes from the generated combined data can lead to a pure clustering of cells. Table 3 shows the comparisons of the Adjusted Rand Index (ARI) values resulting from the cell clustering. It is evident from the table that features/genes selected from the generated combined data of the LSH-GAN model produce better clustering results than the other competing models. The last column of the table 3 shows the ARI scores of clustering results with the original feature (gene) set.

### Selected genes using LSH-GAN can effectively predict cell clusters

Here we provide the detailed results of clustering on four used datasets using the genes selected from the generated samples. For this, we adopted a widely used single cell clustering method SC3<sup>28</sup>. Figure 4 panel- A, depicts the t-SNE visualization of predicted clusters and their original labels for Yan and Pollen datasets (see supplementary figure-1 for the results of the other two datasets). Panel-B of figure 4 represents heatmaps of cell × cell consensus matrix. Each heatmap signifies the number of times a pair of cells is appearing in the same cluster<sup>28</sup>. Here two cells are said to be in different clusters if the score is zero (blue color). Similarly, a score ‘1’ (red) signifies two cells are belonging to the same class. Thus completely red diagonals and blue off-diagonals represent a perfect clustering. A careful notice on the figure 4, panel-A and -B reveals a perfect match between the original and predicted labels for YAN and Pollen datasets.



**Figure 4.** Figure shows the clustering results of Pollen and Yan data sets. Panel-A shows the t-SNE visualization of clustering results (original and predicted labels), whereas panel-B shows the consensus clustering plots of obtained clusters.

## Discussions

In this paper, we present a novel and faster way of generating cell samples of HDSS single cell RNA-seq data using a generative model called LSH-GAN. We update the training procedure of generative adversarial network (GAN) using locality sensitive hashing which can produce realistic samples in a lesser number of iteration than the traditional GAN model. We utilized the generated data in the standard procedure of downstream analysis for analyzing real life scRNA-seq data. Particularly, we demonstrated that the recent and benchmark approaches of gene selection and cell clustering produce excellent results on the generated cell samples of LSH-GAN. Our preliminary simulation experiment also suggests that for fixed number of training iteration the proposed model can generate more realistic samples than the traditional GAN model. This observation is also established theoretically by proving that the cost of value function is less than  $-\log 4$  which is the cost for traditional GAN at the global minimum of virtual training criterion ( $p_g = p_{data}$ ).

We demonstrated that the generated samples of LSH-GAN are useful for gene selection and cell clustering in HDSS scRNA-seq data. particularly the excellent results of LSH-GAN over the recent benchmark methods support its usability for generating realistic cell samples. For validation of the generated cell samples, we use the conventional steps of downstream analysis for scRNA-seq data. We employ five widely used gene selection techniques and one single cell clustering technique for gene selection and grouping of cells. The precise clustering of cells demonstrates the quality of generated cell samples using the LSH-GAN model.

One limitation of our method is that for feature selection we hardly found any linear relationship between the clustering results with the sample size of generated scRNA-seq data. The correct sample size should be selected by using a different range of values between  $0.25p$  to  $1.5p$ , where  $p$  is the feature size. There may be some effects of different parameters related to single cell clustering (SC3 method) and feature selection (e.g. different FS methods, number of selected features, etc.) which may play a critical role in the clustering performance. However, we found clustering results are always better for the generated data with more than  $1p$  ( $p$  is the feature size) sample size. This observation suggests that for feature selection in HDSS data, whenever we produce samples larger than the feature size we will end up with a better clustering. The feasibility of generating such samples is justified by the faster training procedure of LSH-GAN model.

It may interesting to speculate how well LSH-GAN can be useful for generating data of other biological domains, particularly data generated from spatial transcriptomics. The obtained data from this technology has spatial arrangement of cell types within a tissue and thus extremely useful to understand normal development and disease pathology. In-silico generation of this data may find great interest to the machine learning researcher as the model should capture the location-wise heterogeneity of the

real samples.

Taken together, the proposed model can generate good quality cell samples from HDSS scRNA-seq data in a lesser number of iteration than the traditional GAN model. Results show that LSH-GAN not only leads over the benchmarks in the cell sample generation of scRNA-seq data but also accelerates the way of gene selection and cell clustering in the downstream analysis. We believe that LSH-GAN may be an important tool for computational biologists to explore the realistic cell samples of HDSS scRNA-seq data and its application in the downstream analysis.

## Materials and method

### Overview of Datasets

We have used five public benchmark scRNA-seq datasets: Melanoma<sup>31</sup> Pollen<sup>32</sup>, Darmanis<sup>33</sup>, Yan<sup>34</sup> and Klein<sup>35</sup> downloaded from Gene Expression Omnibus (GEO) <https://www.ncbi.nlm.nih.gov/geo/>. Table 4 shows a detailed summary of the used datasets (see supplement for description). The sample:feature ratio for all the datasets except Melanoma are less than 0.012. For Melanoma the ratio is quite large (3.41). We retain this data to know the efficacy of our model in both small and large sample data.

**Table 4.** A brief summary of the datasets used in the experiments.

# Serial	Dataset Name	Features	Instances	Class
1	Yan <sup>34</sup>	20214	90	7
2	Klein <sup>35</sup>	24175	2717	4
3	Darmanis <sup>33</sup>	22088	466	9
4	Pollen <sup>32</sup>	23794	299	11
5	Melanoma <sup>31</sup>	19783	68579	14

### Data Preprocessing

The raw count matrix  $M \in \mathbb{R}^{c \times g}$ , where  $c$  and  $g$  represents the number of cells and genes, respectively, is normalized using *Linnorm*<sup>36</sup> Bioconductor package of R. We select cells having more than a thousand expressed genes (non zero values) and choose genes having a minimum read count more than 5 in at least 10% of the cells.  $\log_2$  normalization is performed on the transformed matrix by adding one as a pseudo count.

### Formal details of LSH-GAN

In this section, we first provide a short description of Generative Adversarial Network (GAN) and then explain the theoretical foundation of LSH-GAN model.

**Generative adversarial network** Generative adversarial network (GAN) is introduced in<sup>17</sup> which was proposed to train a generative model. GAN consists of two blocks: a generative model ( $G$ ) that learn the data distribution ( $p(x)$ ), and a discriminative model ( $D$ ) that estimates the probability that a sample came from the training data ( $X$ ) rather than from the generator ( $G$ ). These two models can be non-linear mapping functions such as two neural networks. To learn the generator distribution  $p_g$  over data  $x$ , a differentiable mapping function is built by generator  $G$  to map a prior noise distribution  $p_z(z)$  to the data space as  $G(z; \theta_g)$ . The discriminator function  $D(x; \theta_d)$  returns a single scalar that represents the probability of  $x$  coming from the real data rather than from generator distribution  $p_g$ . The goal of the generator is to fool the discriminator, which tries to distinguish between true and generated data. Training of  $D$  ensures that the discriminator can properly distinguish samples coming from both training samples and the generator.  $G$  and  $D$  are simultaneously trained to minimize  $\log(1 - D(G(z)))$  for  $G$  and maximize  $\log(D(x))$  for  $D$ . It forms a two-player min-max game with value function  $V(G, D)$

$$\min_G \max_D V(G, D) = E_{x \sim p_x(x)} [\log(D(x))] + E_{z \sim p_z(z)} [1 - \log(D(G(z)))] \quad (2)$$

**Locality Sensitive Hashing Generative adversarial network** For LSH-GAN, a sub-sampling of real data  $p_{x_s}(x_s)$  is augmented with the prior noise distribution,  $p_z(z)$ . Due to this additional information in generator, we assume that the probability  $D(G(\hat{z}))$  will increase by a factor,  $\zeta$ .

the value function of LSH-GAN can be written as:

$$L(D, G) = \min_G \max_D (E_{x \sim p_{data}(x)} \log(D(x)) + E_{\hat{z} \sim p_{\hat{z}}(\hat{z})} \log(1 - D(G(\hat{z})))) \quad (3)$$

**Proposition 1:**  $L(D, G)$  is maximized with respect to Discriminator ( $D$ ), for a fixed Generator ( $G$ ), when

$$D_G^*(x) = \frac{p_{data}(x)(1 - \zeta)}{p_{data}(x) + p_g(x)} \quad (4)$$

**Proof.** Equation 3 can be written as

$$\begin{aligned} L(D, G) &= \int_x p_{data}(x) \log(D(x)) dx + \int_{\tilde{z}} p_{\tilde{z}}(\tilde{z}) \log(1 - \{D(G(\tilde{z})) + \zeta\}) d\tilde{z} \\ &= \int_x p_{data}(x) \log(D(x)) + p_g(x) \log(1 - \{D(x) + \zeta\}) dx \end{aligned} \quad (5)$$

[As the range of  $D(G(\tilde{z}))$  is within the domain of real data  $x$  so we can write this]

We know that, the function  $y = a \log x + b \log(1 - (x + \zeta))$  will have maximum value, at  $x = \frac{a(1-\zeta)}{a+b}$ , for any  $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$  and  $\zeta \in (0, 1)$ . So, the optimum value of  $D$  for a fixed generator,  $G$  is:

$$D_G^*(x) = \frac{p_{data}(x)(1 - \zeta)}{p_{data}(x) + p_g(x)} \quad (6)$$

The training objective for discriminator  $D$  is to maximize the log-likelihood of the conditional probability  $P(Y = y|x)$ , where  $Y$  signify whether  $x$  is coming from real data distribution ( $y = 1$ ) or coming from the generator ( $y = 0$ ). Now the equation 3 can be written as

$$\begin{aligned} C(G) &= \max_D L(G, D) \\ &= (E_{x \sim p_{data}(x)} \log(D_G^*(x)) + E_{\tilde{z} \sim p_g(\tilde{z})} \log(1 - D_G^*(G(\tilde{z}))) \\ &= (E_{x \sim p_{data}(x)} \log(D_G^*(x)) + E_{x \sim p_g(x)} \log(1 - D_G^*(x))) \\ &= E_{x \sim p_{data}(x)} \log \frac{p_{data}(x)(1 - \zeta)}{p_{data}(x) + p_g(x)} + E_{x \sim p_g(x)} \log \left( 1 - \frac{p_{data}(x)(1 - \zeta)}{p_{data}(x) + p_g(x)} \right) \end{aligned} \quad (7)$$

**Theorem 1:** At  $p_g(x) = p_{data}(x)$  (global minimum criterion of value function  $L(G, D)$ ), the value of  $C(G)$  is less than  $(-\log 4)$ .

**proof** From equation 7 we get

$$\begin{aligned} C(G) &= E_{x \sim p_{data}(x)} \log \left( \frac{p_{data}(x)(1 - \zeta)}{p_{data}(x) + p_g(x)} \right) + E_{x \sim p_g(x)} \log \left( 1 - \frac{p_{data}(x)(1 - \zeta)}{p_{data}(x) + p_g(x)} \right) \\ &= E_{x \sim p_{data}(x)} \log \left( \frac{p_{data}(x)(1 - \zeta)}{p_{data}(x) + p_g(x)} \right) + E_{x \sim p_g(x)} \log \left( \frac{\zeta p_{data}(x) + p_g(x)}{p_{data}(x) + p_g(x)} \right) \\ &= \left[ \log(1 - \zeta) + E_{x \sim p_{data}(x)} \log \left( \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) \right] + \left[ E_{x \sim p_g(x)} \log \left( 1 + \frac{\zeta p_{data}(x)}{p_g(x)} \right) + E_{x \sim p_g(x)} \log \left( \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right] \\ &= \left[ \log(1 - \zeta) + E_{x \sim p_g(x)} \log \left( 1 + \frac{\zeta p_{data}(x)}{p_g(x)} \right) \right] + \left[ E_{x \sim p_{data}(x)} \log \left( \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) + E_{x \sim p_g(x)} \log \left( \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right] \\ &= \left[ \log(1 - \zeta) + E_{x \sim p_g(x)} \log \left( 1 + \frac{\zeta p_{data}(x)}{p_g(x)} \right) \right] + [(-\log 4) + 2JSD(p_{data}(x) || p_g(x))] \end{aligned} \quad (8)$$

where,  $JSD(p_{data}(x) || p_g(x))$  represents Jensen–Shannon divergence between two distributions  $p_{data}$  and  $p_g$ . Now, if the two distribution are equal, Jensen–Shannon divergence (JSD) will be zero. Thus, for global minimum criterion of the value function ( $p_g = p_{data}$ ) the Equation 8 is reduces to,

$$C(G) = \log(1 - \zeta) + \log(1 + \zeta) + (-\log 4) = \log \frac{(1 - \zeta^2)}{4} \leq (-\log 4) \quad (9)$$

This completes the proof.

## Competing interests

The authors declare that they have no competing interests.

## Author's contributions

SL and SR equally contributed to this work

## Acknowledgment

We would like to acknowledge support from J.C. Bose Fellowship [SB/S1/JCB- 033/2016 to S.B.] by the DST, Govt. of India; SyMeC Project grant [BT/Med-II/NIBMG/SyMeC/2014/Vol. II] given to the Indian Statistical Institute by the Department of Biotechnology (DBT), Govt. of India. Govt. of India; Inspire DST Project.

## References

1. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell rna sequencing data: challenges and opportunities. *Nat. methods* **14**, 565 (2017).
2. Ray, S. & Schonhuth, A. Markercapsule: Explainable single cell typing using capsule networks. *bioRxiv* (2020).
3. Liao, S., Gao, Q., Nie, F., Liu, Y. & Zhang, X. Worst-case discriminative feature selection. In *IJCAI*, 2973–2979 (2019).
4. Duò, A., Robinson, M. D. & Soneson, C. A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research* **7** (2018).
5. Luecken, M. D. & Theis, F. J. Current best practices in single-cell rna-seq analysis: a tutorial. *Mol. systems biology* **15**, e8746 (2019).
6. Ji, Z. & Ji, H. Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic acids research* **44**, e117–e117 (2016).
7. Plass, M. *et al.* Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science* **360** (2018).
8. Fincher, C. T., Wurtzel, O., de Hoog, T., Kravarik, K. M. & Reddien, P. W. Cell type transcriptome atlas for the planarian *schmidtea mediterranea*. *Science* **360** (2018).
9. Chen, G., Ning, B. & Shi, T. Single-cell rna-seq technologies and related computational data analysis. *Front. genetics* **10**, 317 (2019).
10. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. biotechnology* **36**, 411–420 (2018).
11. Vans, E., Patil, A. & Sharma, A. Feats: Feature selection based clustering of single-cell rna-seq data. *bioRxiv* (2020).
12. Lall, S., Sinha, D., Bandyopadhyay, S. & Sengupta, D. Structure-aware principal component analysis for single-cell rna-seq data. *J. Comput. Biol.* (2018).
13. Lall, S., Ray, S. & Bandyopadhyay, S. Rgcop-a regularized copula based method for gene selection in single cell rna-seq data. *bioRxiv* (2020).
14. Marouf, M., Machart, P., Bansal, S., Bonn *et al.* Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks. *Nat. communications* **11**, 1–12 (2020).
15. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell rna sequencing data. *Genome biology* **18**, 1–15 (2017).
16. Lindenbaum, O., Stanley, J., Wolf, G. & Krishnaswamy, S. Geometry based data generation. In Bengio, S. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 31 (Curran Associates, Inc., 2018).
17. Goodfellow, I. *et al.* Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680 (2014).
18. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, 214–223 (PMLR, 2017).
19. Nowozin, S., Cseke, B. & Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709* (2016).

20. Karras, T., Aila, T., Laine, S. & Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations* (2018).
21. Mirza, M. & Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
22. Pauleve, L., Jegou, H. & Amsaleg, L. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognit. Lett.* **31**, 1348–1358 (2010).
23. Mao, X.-L. *et al.* S2jsh-lsh: A locality-sensitive hashing schema for probability distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31 (2017).
24. Townes, F. W., Hicks, S. C., Aryee, M. J. & Irizarry, R. A. Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. *Genome biology* **20**, 1–16 (2019).
25. Andrews, T. S. & Hemberg, M. M3drop: dropout-based feature selection for scrnaseq. *Bioinformatics* **35**, 2865–2867 (2019).
26. Grün, D., Kester, L. & Van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nat. methods* **11**, 637 (2014).
27. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* DOI: [10.1016/j.cell.2021.04.048](https://doi.org/10.1016/j.cell.2021.04.048) (2021).
28. Kiselev, V. Y. *et al.* Sc3: consensus clustering of single-cell rna-seq data. *Nat. methods* **14**, 483–486 (2017).
29. Yang, L.-C. & Lerch, A. On the evaluation of generative models in music. *Neural Comput. Appl.* **32**, 4773–4784 (2020).
30. Lucic, M., Kurach, K., Michalski, M., Gelly, S. & Bousquet, O. Are gans created equal? a large-scale study. *Adv. Neural Inf. Process. Syst.* (2017).
31. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science* **352**, 189–196 (2016).
32. Pollen, A. A. *et al.* Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat. biotechnology* **32**, 1053 (2014).
33. Darmanis, S. *et al.* A survey of human brain transcriptome diversity at the single cell level. *Proc. Natl. Acad. Sci.* **112**, 7285–7290 (2015).
34. Yan, L. *et al.* Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nat. structural & molecular biology* **20**, 1131 (2013).
35. Klein, A. M. *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **161**, 1187–1201 (2015).
36. Yip, S. H., Wang, P., Kocher, J.-P. A., Sham, P. C. & Wang, J. Linnorm: improved statistical analysis for single cell rna-seq expression data. *Nucleic acids research* **45**, e179–e179 (2017).

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [supplementaryNMIupdated.pdf](#)