Protocol

A deep learning framework for denoising and ordering scRNA-seq data using adversarial autoencoder with dynamic batching



Single-cell RNA sequencing (scRNA-seq) provides high resolution of cell-to-cell variation in gene expression and offers insights into cell heterogeneity, differentiating dynamics, and disease mechanisms. However, technical challenges such as low capture rates and dropout events can introduce noise in data analysis. Here, we present a deep learning framework, called the dynamic batching adversarial autoencoder (DB-AAE), for denoising scRNA-seq datasets. First, we describe steps to set up the computing environment, training, and tuning. Then, we depict the visualization of the denoising results.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Kyung Dae Ko, Vittorio Sartorelli

vittorio.sartorelli@nih.gov

CellPress

Highlights

Steps for denoising scRNA-seq using dynamic batching adversarial autoencoder (DB-AAE)

Instructions to train DB-AAE and visualize denoising results

Guidance on inferring pseudotime from the denoising results

Ko & Sartorelli, STAR Protocols 5, 103067 June 21, 2024 Published by Elsevier Inc. https://doi.org/10.1016/ j.xpro.2024.103067

Protocol



A deep learning framework for denoising and ordering scRNA-seq data using adversarial autoencoder with dynamic batching

Kyung Dae Ko^{1,2} and Vittorio Sartorelli^{1,3,*}

¹Laboratory of Muscle Stem Cells and Gene Regulation, NIAMS, NIH, Bethesda, MD, USA ²Technical contact ³Lead contact

*Correspondence: vittorio.sartorelli@nih.gov https://doi.org/10.1016/j.xpro.2024.103067

SUMMARY

Single-cell RNA sequencing (scRNA-seq) provides high resolution of cell-to-cell variation in gene expression and offers insights into cell heterogeneity, differentiating dynamics, and disease mechanisms. However, technical challenges such as low capture rates and dropout events can introduce noise in data analysis. Here, we present a deep learning framework, called the dynamic batching adversarial autoencoder (DB-AAE), for denoising scRNA-seq datasets. First, we describe steps to set up the computing environment, training, and tuning. Then, we depict the visualization of the denoising results.

For complete details on the use and execution of this protocol, please refer to Ko et al.¹

BEFORE YOU BEGIN

Deep learning techniques, particularly autoencoders, have been developed to address noise reduction and missing value imputation challenges in single-cell RNA sequencing (scRNA-seq) data analysis. However, these methods can exhibit sensitivity to sparse data and batch effects. To improve denoising performance and alleviate information loss during analysis, we propose an approach called the dynamic batching adversarial autoencoder (DB-AAE). This model utilizes a competitive architecture that directly captures optimal features from input data, bypassing the need for traditional statistical models.

The DB-AAE comprises three crucial components: encoder, decoder, and adversary modules (Figure 1). This protocol provides comprehensive step-by-step instructions to denoise scRNA-seq data using the DB-AAE framework, suitable for both beginners and seasoned professionals.

For complete details on the use and execution of this protocol, please refer to Ko et al.¹

Hardware preparation

The whole framework can run on a MacOS or Window operation system. Even though the RAM requirement depends on the number of cells to be analyzed, 16 GB RAM should be sufficient for an initial analysis.

Software preparation

This framework is based on the Microsoft Windows operating system but it can be run on MacOS. To build a functional Python environment, this protocol uses Anaconda v2.5.1 as Anaconda provides various integrated development environment (IDE)s for code development and management.

1







Figure 1. Structure of the dynamic batching adversarial autoencoder (DB-AAE)

Among the IDEs in the Anaconda, JupyterLab IDE v.3.5.3 was selected as it easily visualizes the results through the framework. For additional visualization, R studio v.2023.06.2 + 561 with R v.4.3 was used.

Setting up the computational environment

© Timing: 1 h

- 1. Install Anaconda.
 - a. Download Anaconda from "https://repo.anaconda.com/archive/"
 - b. Install the Anaconda Navigator following the default settings.
 - c. Launch the Anaconda Navigator.
- 2. Set up Virtual environment.
 - a. Create a new environment called to "db_aae" from Environments (Figures 2A and 2B).
 - b. Install JupyterLab from Home after selecting "db_aae" environment (Figure 2C).
 - c. Launch JupyterLab from Home (Figure 2D).
- 3. Install python packages.
 - a. Launch a terminal in JupyterLab (Figure 3A).
 - b. Run the following command in the terminal to install DB-AAE (Figure 3B).

> pip install dbaae

- 4. Install R and R Studio.
 - a. Download R from "https://cloud.r-project.org/" and install it.
 - b. Download R-studio from "https://posit.co/download/rstudio-desktop/" and install it.
- 5. Install R packages.
 - a. Install scater and slingshot.

STAR Protocols Protocol



> if (!require("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install(c("scater", "slingshot"))

b. Install Polychrome.

> install.packages(``Polychrome'')

c. Install zellkonverter.

> if (!require("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("zellkonverter")

d. Install scater.

```
> if (!require("BiocManager", quietly = TRUE))
```

install.packages("BiocManager")

BiocManager::install("scater")

Data collection

© Timing: 30 min

In this section, we collect raw scRNA-seq data from a public archive (https://hemberg-lab.github.io/ scRNA.seq.datasets/mouse/edev/) and convers the data to a Hierarchical Data Format (HDF-.h5ad) for python.

6. Import R libraries.

library("scater")

library(zellkonverter)

7. Convert a dataset to H5ad format and save the dataset.

goolam<-readRDS("../pre_process_data/goolam.rds")</pre>

rowData(goolam)

colData(goolam)

writeH5AD(goolam, file = "goolam.h5ad")



STAR Protocols Protocol

| Α | | | | | в | | | | |
|---|--|---|---|------------|---|---|---|--|-----------|
| () Anaconda Navigator | | | | - 0 × | Q Anaconda Navigator | | | | × |
| | NDA.NAVIGATOR | | O Upgrade Now | Connect V | | DA.NAVIGATOR | | () Upgrade Now | ianaest ~ |
| A Home | (Search Environments Q) | Installed v Channels Update index. | | hPadages Q | A Home | (Search Environments Q) | Institut v Daniels Bodate index. | | (P repert |
| | base (root) | Name w T Description | | vesion * | | have been | Name V T Description | | version m |
| Unvronments | | stell ay This repeatory is a collection of python library code has been extensively tested and used in production. | for building python applications, the code is collected from google's own python code base, and | 1.40 | Environments | 5464 (7000) | 🖬 ebsl-py 👌 This repository is a collection of python library code for build | ing python applications, the code is collected from googla's own python code base, and | 1.4.0 |
| 🖨 Learning | • | 🖬 adjustitent 🗢 | | 0.8 | Learning | | S equatest | | - |
| | 8-300-1 | a statester O Configurable, python 2+3 compatible sphine theme. | | 0.7.12 | - | B-MNR+1 | alabester O Configurable, python 2+3 competible sphine theme. | | 0.7.12 |
| Community | \$29 ₁ erv | 🖬 anndata 🛛 🧔 | | 0.8.0 | Community | SOP_etv | 🖬 avedata 🛛 🐣 | | 0.8.0 |
| | cell2loc,etv | 🛛 avvdetažvi 🔶 | | 10 | | cell2loc ety | 🖬 andal Create new environment 🛛 🗙 🗙 | Type a name of | |
| | omp_autoencorder | anyis O High level compatibility layer for multiple synchron | ous event loop implementations on python | A 330 | | | S myle Name Guard | loop implementations on python | # 3.50 |
| | | S argon2-cf5 O The secure argon2 persiverd hashing algorithm. | | 21.3.0 | | chg_succencorder | argen2 Lécilien Claurini/Anandulandulan | | 21.3.0 |
| | 608 | angon2-off-bindings () Low-level python off-bindings for angon2 | | 21.2.0 | | 60 | 🖬 argong Packaper 🖬 Python 3.5.16 v | | 21.2.0 |
| | dia_test C | B arrow O Better dates & times for python | | 123 | 3 | dca_test | E arrow 0 8 2.41 | | 1.2.3 |
| | predict_binding | astroid O A abstract syntax tree for python with inference supp | ort. | 2.14.2 | | predict_binding | E atroid Cancel Create | | 2.14.2 |
| | sylouth | statiskens O The astrokens module annotates python abstract syn | tax trees (act) with the positions of tokens and text in the source code that generated them. | 205 | | - | E estadem | rests) with the positions of tokens and text in the source code that generated them. | 2.0.5 |
| | n refordate | sourcester An all ungester for pyrion. | | 140 | _ | | 🖬 attunparse 👶 An act unparser for python. | | 143 |
| Anaconda Toelbex | | Server O Attra is the python package that will bring back the jo | y of writing classes by relieving you from the drudgery of implementing object protocols (aka | A 22.5.0 | Anaconda Teelbax | nneticulate | atomicwrites Atomic fue writes | tion shows be realized as one from the designers of inclumenting which exclosely betw | 1.4.0 |
| focul notebooks Click the Toolbex | spetial_transcriptome | dunder methods). | form to the sea 8 style mide | 140 | Supercharged kocol notebooks | spatial_transcriptome | dunder method). | | * 22.1.0 |
| the to matali. | V_analysis 🔘 | tabel Utilities to internationalize and localize python applic | ations . | 2.11.0 | the to match | tf_analysis 🔘 | Accord O Accord according to the second | the pep 8 style puide | 14.0 |
| | zelkonverterknidatafs | backcall O Specifications for caliback functions passed in to an ap | | 02.0 | | zellkonverterikndetadn | Andreil O Secondaria in the sale of a second process of a second secon | | 020 |
| Documentation | | 🛢 backports.zoneinfo 🔹 Backport of the standard library zoneinfo module | | 0.2.1 | Documentation | | 5 hardwards mouthin for Bardward of the standard Drawy mouthin module | | 0.2.1 |
| Anaconda Biog | environment | boyst O Modern password hashing for your software and you | 19740 | 320 | Anaconda Blog | | Intervent O Mostern associated hashing for your software and your strugent | | 120 |
| ¥ 8 9 | | S beautifuloup4 O Python library designed for screen scraping | | A 411.1 - | ¥ & * | | tesutifulacup4 O Python library designed for screen-scraping | | × 411.1 × |
| | Create Clone Import Backup Remove | | | | | Create Clane Import Backup Remove | | | |
| O Anaconda Navigator Elir Isrio O ANACON | NDA.NAVIGATOR | 4011 408 343 | Select specific version | × | | DA.NAVIGATOR | | 🕑 Upgrade Now 💦 🖉 | 0 X |
| ft Home | All applications v on the | # • Channels 352 | | C | 🕈 Home | All applications + on the | Alter V Channels | | C |
| Environments | Destipeli | () () () ()))))) ()))) ())) ()) ()) ()) ()) ()) ()) ()) ()) ()) () | Datalane | | Environments Learning | CataSpell | Jupyterab Jupyterab Netebook | Datalore | Î |
| L Community | Developed is an OE for exploratory data analysis and parotypying module learning models. It combines the interactivity of Jupyier notebools with the interactivity of Jupyier notebools with the interactivity of Python and R colding assistance of Pytharm in one user friendly environment. Install | tendball exervisionen (fr. interactive profile service) and activities of site of profile interactive and activities of the profile interactive and activities of the profile interactive and activities of the profile interactive activities of the profile interactive activities of the profile interactitis of the profile interactive activities of the profile interactive | Kicherster yn det sciener gerigethin Bergener fan e per eerferger en wennen Degener gersteren fer fyngen. Syn Bergener gersteren fer fyngen. Syn | | 🛣 Conmunity | Destgeri i an UE for realizorozy des nardysis and protozglagi mediote learning models. It combines the interactivity of Judgere notebools with the interactivity of proton and it configures the interactivity in one user foreignest (metall) | testing in environment for internation Web-band, internation company The international and and a state of the international and and and and a state of the international and | start y-un diverse services in the service of data realised in a service object week to be each on object realised and be each out out the frage. | |
| | IBM Wattern Studie Cloud IBM Wattern Studie Cloud IBM Wattern Studie Cloud Ibm Hantern Studie Cloud Don't to malying and Visualize data, to clasers and thread data, to cares and train of participation of the studies | Critic but Series Service assessments to 123 and 124 a | CND.exe Prompt 11 In a criteria territoria da pura cameta estatoria da pura cameta 11 11 11 11 11 11 11 11 11 1 | | | IBM Watton Studio Cloud BM Watton Studio Cloud BM Watton Studio Cloud BM Watton Studio Cloud Amount BM Watton Studio Cloud Amount BM Watton BM WAT | Crace Case Server Case Case Server Setter to the two manage and Setter to the number of setter Setter to the number of sets and Setter to the number of sets and Set Sets and Sets and Sets and Set Sets and Sets and Sets and Sets and Sets and Sets and Sets and Sets and Sets and Sets and Sets and Sets and Se | CMLee Point B.1 onderet manufacture automatic Interest manufacture automatic B.1 B.1 B.1 B.1 B.1 B.1 B.1 B.1 | |
| Anseends Tealbex Superviseged local notebooks Class the Tealbook | machine learning models. Frequent data and build models. using parts source data science tools or visual modeling. | 123 104 why have for the service 104 million 104 milli | (Intel) | | Aneconda Techen Supercharged bool netetions tick the toolbox | matche learning models. Prepare data and build models, using consurce data science tools or vigual modeling. | te diada hippode Facella cases source biblio Larendo | (100) | |
| Decumentation Arrested Blog | Glassia Glassia 124 Multidimensional Barkinski School Kina Espine-relationsky within and Barkinski School | Consequence of the second seco | severshell_sheaton_missions. | | Documentation Anaconde Blog | Cleavit 12.4 Multidimensional data subvalization acress Res. Database relationarias within and | Organis Operating for the second | weishell_donnut_minicend 0.01 Akti fleged of by your externin for born Sector (and by types externing). | |
| ¥ ă ? | among related datasets. nov | and expert. Interactive workflows activated 0350 034.12 014.9 | Supports WTMS, JS, and SQL | × | ¥ à ¢ | among resated datasets. No | activated | Supports HTML, JL and SQL | N |

Figure 2. Anaconda Navigator

(A) "Environments" page illustration.

(B) Illustration to create a new conda environment.

(C) "Home" page illustration to install JupyterLab.

(D) Illustration to launch JupyterLab.

Note: Single-cell datasets analyzed in this protocol were preprocessed and deposited into Google Drive. The link is represented in GitHub (https://github.com/LMSCGR/DB-AAE).

KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER | | |
|----------------------------------|---|--|--|--|
| Deposited data | | | | |
| goolam | Goolam et al. ² | https://drive.google.com/drive/u/1/folders/ 1DwRpIEStBTQ7XfAeI7Qq1uJeCRZyUj9U | | |
| baron | Baron et al. ³ | https://drive.google.com/drive/u/1/folders/ 1DwRpIEStBTQ7XfAel7Qq1uJeCRZyUj9U | | |
| Software and algorithms | | | | |
| R v.4.3 | The R Project for Statistical Computing | https://www.r-project.org/ | | |
| RStudio server (v.2023.06.2+561) | RStudio Team, 2022 | https://posit.co/ | | |
| Python v.3.8 | Python Software Foundation | https://cran.r-project.org/web/packages/Seurat/index.html | | |
| scanpy v.1.9.1 | Wolf et al. ⁴ | https://github.com/scverse/scanpy | | |
| seaborn v.0.12.2 | Waskom ⁵ | https://github.com/mwaskom/seaborn | | |
| pandas v.1.5.3 | McKinney ⁶ | https://github.com/pandas-dev/pandas | | |
| matplotlib v.3.6.3 | Hunter ⁷ | https://github.com/matplotlib/matplotlib | | |
| numpy v.1.23.5 | Harris et al. ⁸ | https://github.com/numpy/numpy | | |

Protocol



| Continued | | | | |
|---|----------------------------|---|--|--|
| REAGENT or RESOURCE | SOURCE | IDENTIFIER | | |
| TensorFlow v.2.11.0 | TensorFlow team, 2015 | https://www.tensorflow.org/ | | |
| keras-tuner v.1.4.5 | | https://github.com/keras-team/keras-tuner | | |
| keras v.2.11.0 Keras team, 2015 | | https://github.com/fchollet/keras | | |
| zellkonverter v.1.12.1 | | https://github.com/theislab/zellkonverter | | |
| Polychrome v.1.5.1 | | https://cran.r-project.org/web/packages/Polychrome/index.html | | |
| Slingshot v.2.10.0 | Street et al. ⁹ | https://github.com/kstreet13/slingshot | | |
| scater v.1.28.0 McCarthy et al. ¹⁰ | | http://bioconductor.org/packages/scater/ | | |

STEP-BY-STEP METHOD DETAILS

In this section, we describe essential steps to train DB-AAE (Dynamic batching adversarial auto encoder).

Step 1: Training DB-AAE

© Timing: 1 h



Figure 3. Setting up the computational environment

(A) JupyterLab illustration.

- (B) Illustration to run a terminal for installation of DB-AAE package.
- (C) Illustration to launch a Jupyter notebook.

(D) Illustration to run python codes in a Jupyter notebook.

1. Create jupyter notebook (Figure 3C).





2. Import python package.

| from dbaae.utils import train | |
|-------------------------------|--|
| from dbaae.utils import io | |
| import random | |
| import time | |
| import datetime | |
| import pandas as pd | |

3. Load a dataset.

| random.seed(1234) |
|---|
| input_path="./data/goolam.h5ad" |
| genenum=5000 |
| latent_dim=512 |
| adata=io.read_dataset(input_path,highly_genes=highly_genes) |

4. Setup hyper parameters.

| input_layer=genenum |
|---|
| hidden_layer1=1024 |
| hidden_layer2=512 |
| latent_layer=latent_dim |
| batch_size = 32 |
| n_epochs = 25 |
| X_train = pd.DataFrame(adata.X) |
| <pre>X_train = X_train.to_numpy()</pre> |
| activation='relu' |
| optimizer="RMSprop" |
| learningrate=2e-05 |
| <pre>start_time = time.time()</pre> |

Note: To use a specific dataset, the "input_path" can be changed to the location with the specific filename. If the accuracy after step 5 is not high, "optimizer" and "learningrate" can be changed following guide from keras website (https://keras.io/api/optimizers/).

5. Train DB-AAE.





a. Train with static batching (or).

```
adversarial_autoencoder=train.train(X_train,input_layer,hidden_layer1,hidden_layer2,latent_layer,batch_size,
n_epochs,learningrate,activation,optimizer)
```

adversarial_autoencoder.summary()

b. Train with dynamic batching.

from dbaae.utils import dynamic import keras_tuner as kt import tensorflow as tf tuner = kt.Hyperband(dynamic.AEHyperModel(X_train,input_layer,hidden_layer1,hidden_layer2,latent_layer, n_epochs,learningrate,activation,optimizer), objective = 'val_accuracy', max_epochs = n_epochs, executions_per_trial = 1, overwrite = True, directory = './cache', max_consecutive_failed_trials=5, factor = 3) stop_early = tf.keras.callbacks.EarlyStopping(monitor='val_accuracy', mode='max', baseline=100) tuner.search(X_train, X_train, epochs = n_epochs, validation_split = 0.1, callbacks=[stop_early]) best_hps = tuner.get_best_hyperparameters(num_trials=1)[0] print(f"""batch_size: {best_hps.get('batch_size')}""") batch_size = best_hps.get('batch_size') random.seed(1234) adversarial_autoencoder=train.train(X_train, input_layer, hidden_layer1, hidden_layer2, latent_layer, batch_size, n_epochs, learningrate, activation, optimizer)

adversarial_autoencoder.summary()

▲ CRITICAL: Step 5-a describes static batching and step-b is dynamic batching. In step 5-b, to use dynamic batching for training, "n_echochs" can be changed to small number to reduce training time since the training time of dynamic batching is 2–3 time slower than that of static batching. Follow guidelines from keras website (https://keras.io/api/keras_tuner/tuners/hyperband/).

6. Save the trained model.

```
rlt_dir='./models'
current_time = datetime.datetime.now()
adversarial_autoencoder.save(rlt_dir+'{}'.format("DB_AAE")+'_'+-
str(current_time.year)+'_'+str(current_time.month)+'_'+
str(current_time.day)+'_'+str(current_time.hour)+'_'+str(current_time.minute)+'.h5')
```





Step 2: Visualize the clusters of the denoised scRNA-seq dataset

© Timing: 15 min

7. Import the trained model.

```
import keras
```

db_aae = keras.models.load_model('./model/DB_AAE_2024_2_20_10_31.h5')

8. Load original data.

| random.seed(1234) |
|--|
| input_path="./data/goolam.h5ad" |
| genenum=5000 |
| latent_dim=512 |
| adata=io.read_dataset(input_path,highly_genes=genenum) |

9. Reconstruct the data.

| X_train = pd.DataFrame(adata.X) |
|---|
| X_train=X_train.to_numpy() |
| cellinfo=pd.DataFrame(adata.obs['cell_type1']) |
| <pre>geneinfo=pd.DataFrame(adata.var['feature_symbol'])</pre> |
| <pre>counts_aae = db_aae.predict(X_train)</pre> |
| import scanpy as sc |
| adata_aae = sc.AnnData(counts_aae[0],obs=cellinfo,var=geneinfo) |
| adata_aae.write("./results/DB_AAE_2024_2_20_10_31.h5ad") |

10. Preprocess the data for visualization of clusters using SCANPY.

| sc.pp.normalize_per_cell(adata_aae) |
|---|
| <pre>sc.pp.log1p(adata_aae)</pre> |
| sc.pp.pca(adata_aae) |
| <pre>sc.pl.pca_variance_ratio(adata_aae)</pre> |
| <pre>sc.pp.neighbors(adata_aae, n_neighbors=60, n_pcs=10)</pre> |

11. Visualize the clusters (Figure 4A).

```
sc.tl.umap(adata_aae)
```

sc.pl.umap(adata_aae, color='cell_type1',title='Reconstructed data')

Protocol





Figure 4. Visualization of a denoised scRNA-seq dataset(A) UMAP plot of a denoised scRNA-seq dataset.

(B) Plot to infer pseudo time of a denoised scRNA-seq dataset.

Note: "n_neighbors" controls how UMAP balances local versus global structure in the data and "n_pcs" decides the number of PCs involved in clustering. Their numbers are decided based on "pca_variance_ratio" plot in step 10. After the value of n_neighbors that makes "pca_variance_ratio" plot smooth is selected, the point that starts to smoothen and flatten out in the plot is chosen for the value of n_pcs.

Step 3: Infer pseudotime of the denoised scRNA-seq dataset

© Timing: 15 min

- 12. Infer pseudotime from the denoised dataset using R.
 - a. Import R libraries.

library("slingshot")
library("scater")
library(zellkonverter)
library(ggplot2)
library(ggbeeswarm)
library(ggthemes)
library(Polychrome)





b. Load a dataset.

```
aae_goolam_sce<-readH5AD("./results/DB_AAE_2024_2_20_10_31.h5ad")
logcounts(aae_goolam_sce)<-assay(aae_goolam_sce)</pre>
```

c. Preprocess a dataset for the visualization of pseudo time.

| aae_goolam_sce@colData\$cell_type1<-ordered(aae_goolam_sce@colData\$cell_type1, levels = c("2cell", "4cell", "8cell", "16cell", "blast")) | |
|--|--|
| <pre>cell.stages= c("2cell", "4cell", "8cell", "16cell", "blast")</pre> | |
| <pre>label=ordered(aae_goolam_sce\$cell_type1, levels = c("2cell", "4cell","8cell","16cell","blast"))</pre> | |
| set.seed(723451) | |
| <pre>my_color <- createPalette(10, c("#010101", "#ff0000"), M=1000)</pre> | |
| <pre>names(my_color) <- unique(as.character(aae_goolam_sce\$cell_type1))</pre> | |
| <pre>aae_goolam_sce <- runPCA(aae_goolam_sce, ncomponents = 20)</pre> | |
| aae_goolam_sce <- runUMAP(aae_goolam_sce) | |
| aae_goolam_sce <- slingshot(aae_goolam_sce,reducedDim = "PCA") | |
| slingshot_df <- data.frame(slingPseudotime_1=aae_goolam_sce@colData\$slingPseudotime_ cell_type=aae_goolam_sce@colData\$cell_type1) | |
| <pre>slingshot_df\$cell_type <- ordered(slingshot_df\$cell_type, levels = c("2cell", "4cell","8cell","16cell","blast"))</pre> | |
| set.seed(723451) # for reproducibility | |
| <pre>my_color <- createPalette(10, c("#010101", "#ff0000"), M=1000)</pre> | |
| <pre>names(my_color) <- unique(as.character(aae_goolam_sce\$cell_type1))</pre> | |
| aae_goolam_sce <- runPCA(aae_goolam_sce, ncomponents = 20) | |
| aae_goolam_sce <- runUMAP(aae_goolam_sce) | |
| aae_goolam_sce <- slingshot(aae_goolam_sce,reducedDim = "PCA") | |
| slingshot_df <- data.frame(slingPseudotime_1=aae_goolam_sce@colData\$slingPseudotime_1, cell_type=aae_goolam_sce@colData\$cell_type1) | |
| <pre>slingshot_df\$cell_type <- ordered(slingshot_df\$cell_type, levels = c("2cell", "4cell", "8cell","16cell","blast"))</pre> | |

d. Visualize the pseudotime (Figure 4B).

STAR Protocols Protocol



xlab("Pseudotime") + ylab("cell type") +

scale_colour_manual(values = my_color)+theme(text = element_text(size = 20))

EXPECTED OUTCOMES

This protocol outlines a generative framework named DB-AAE designed to tackle the inherent challenges related to denoising and imputation in single-cell RNA sequencing (scRNA-seq) data. By employing a dynamic batching procedure in step 1 during the training of DB-AAE, an optimized model with an ideal batch size can be constructed. The trained model can generate a denoised scRNA-seq dataset in h5ad format. Subsequently, in step 2, the denoised dataset can be visualized, and its pseudo-time can be inferred.

LIMITATIONS

As the implementation of DB-AAE relies on a deep-learning model, this framework faces limitations in offering detailed insights into the specific acquisition and utilization of features or gene expression patterns by the model. This constraint arises from the intricate nature of deep learning models, characterized by numerous layers with complex interactions among nodes. Consequently, the internal workings or processes of the model are challenging to comprehend or interpret. Moreover, the performance of the DB-AAE framework is highly sensitive to hyperparameters, necessitating thorough hyperparameter tuning for ensuring the method's stability and robustness.

TROUBLESHOOTING

Problem 1 HTTP error when installing Python packages (related to Step 3).

Potential solution

This is due to the poor quality of internet connection to the package sources. A possible solution to solve this problem is to check internet connection. If the connection is recovered, repeat steps in Figures 2C, 2D, and 3A.

Problem 2

IPython Console reports an error of "No module named XX" when executing the codes (related to Step 3).

Potential solution

• This is because the correct package version is not installed. Open the terminal (Figure 3A) and check the installed package with this command.

conda list

• Check python packages in GitHub (https://github.com/LMSCGR/DB-AAE) and reinstall the packages with this command in the terminal.

pip install package name == version





Problem 3

R packages cannot be loaded by "library" command (related to Step 6).

Potential solution

Run the codes in R environment below.

```
p <- installed.packages()
colnames(p)</pre>
```

If the packages cannot be found after running the codes, visit the Bioconductor website (https:// www.bioconductor.org/), search a package, and follow guidelines. If the package cannot be found in Bioconductor, run install.packages("package_name") in R environment.

Problem 4

Data files cannot be loaded (related to Step 7).

Potential solution

Check whether the files are in the folder. If they are, check their name.

Problem 5

Plots cannot be drawn (related to Step 11 and Step 12f).

Potential solution

Run the codes in R environment below.

```
gg2 <- try(find.package("ggplot2"), silent = TRUE)</pre>
```

gg2

If the packages cannot be found after running the codes, run **install.packages("ggplot2"**) in R environment.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Vittorio Sartorelli (vittorio.sartorelli@nih.gov).

Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contact, Kyung Dae Ko (kyungdae.ko@nih.gov).

Materials availability

This study did not generate unique reagents.

Data and code availability

Original data and codes have been deposited to Zenodo: https://doi.org/10.5281/zenodo.10478925.

ACKNOWLEDGMENTS

We thank members of the Sartorelli's laboratory for insightful comments. Funding has been provided by NIH grants AR041126 and AR041164.

Protocol



AUTHOR CONTRIBUTIONS

K.D.K. analyzed and interpreted data and drafted the manuscript. V.S. edited the manuscript and supervised the project.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

- Ko, K.D., and Sartorelli, V. (2024). A deep learning adversarial autoencoder with dynamic batching displays high performance in denoising and ordering scRNA-seq data. iScience 27, 109027.
- Goolam, M., Scialdone, A., Graham, S.J.L., Macaulay, I.C., Jedrusik, A., Hupalowska, A., Voet, T., Marioni, J.C., and Zernicka-Goetz, M. (2016). Heterogeneity in Oct4 and Sox2 Targets Biases Cell Fate in 4-Cell Mouse Embryos. Cell 165, 61–74.
- Baron, M., Veres, A., Wolock, S.L., Faust, A.L., Gaujoux, R., Vetere, A., Ryu, J.H., Wagner, B.K., Shen-Orr, S.S., Klein, A.M., et al. (2016). A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-

cell Population Structure. Cell Syst. 3, 346–360.e4.

- 4. Wolf, F.A., Angerer, P., and Theis, F.J. (2018). SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 19, 15.
- 5. Waskom, M. (2021). seaborn: statistical data visualization. J. Open Source Softw. 6, 3021.
- 6. McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, 445.
- 7. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D.,

Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array programming with NumPy. Nature 585, 357–362.

- 8. Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Comput. Sci. Eng. 9, 90–95.
- Street, K., Risso, D., Fletcher, R.B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S. (2018). Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genom. 19, 477.
- McCarthy, D.J., Campbell, K.R., Lun, A.T.L., and Wills, Q.F. (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. Bioinformatics 33, 1179–1186.