
MCBRLab
bilibili
superbio
Berke · Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai

Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke · Superbio.ai


MCBRLab
bilibili
Haotian Cui


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
Haotian Cui


MCBRLab
bilibili
Haotian Cui


MCBRLab
bilibili
Haotian Cui


MCBRLab
bilibili
Haotian Cui


MCBRLab
bilibili
Haotian Cui


MCBRLab
uperbio
Berke · Superbio.ai

MCBRLab bilibili

scGPT: towards building a foundation model for single-cell multi-omics using generative AI

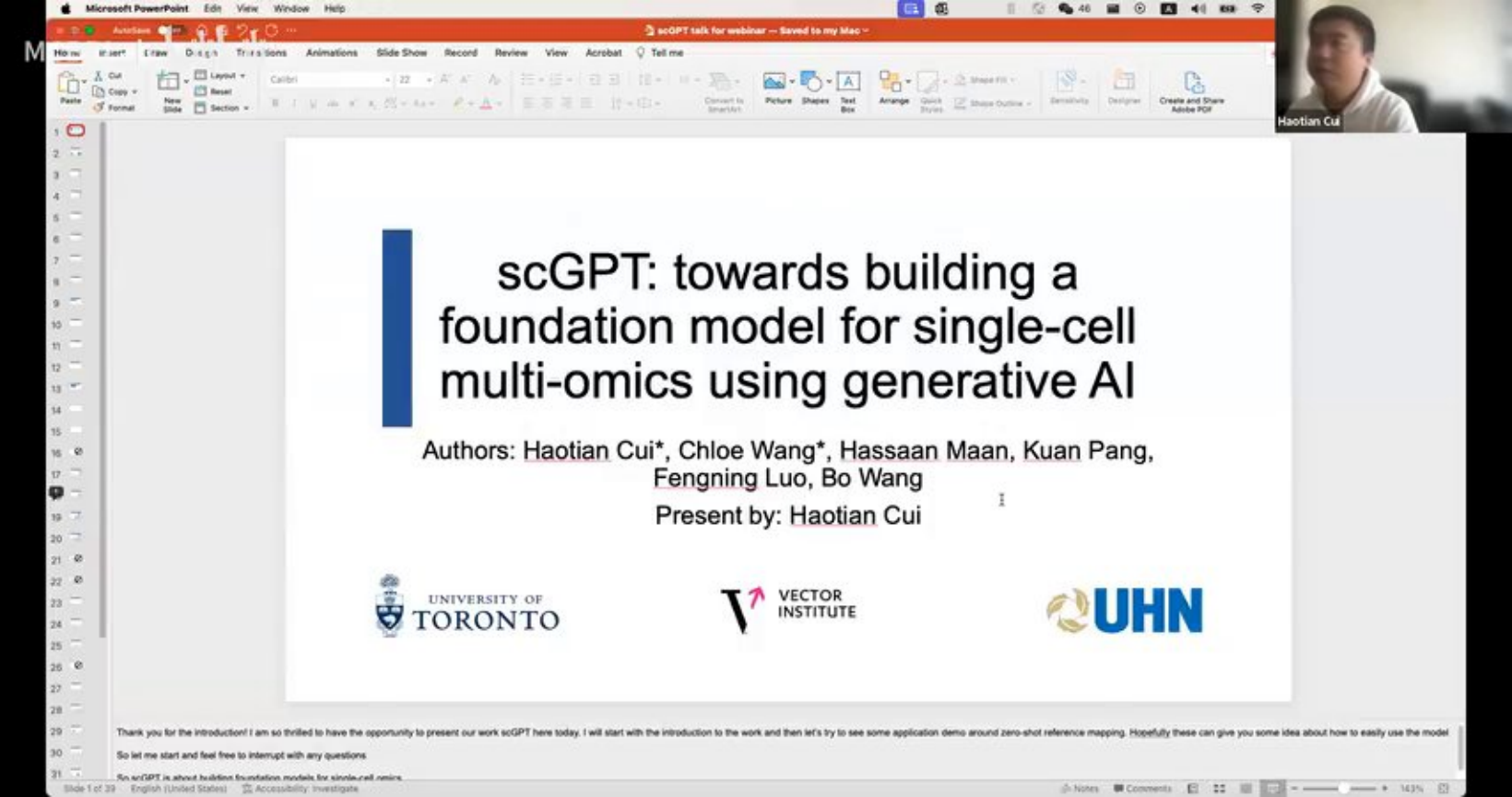
Authors: Haotian Cui*, Chloe Wang*, Hassaan Maan, Kuan Pang, Fengning Luo, Bo Wang

Present by: Haotian Cui

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

UHN


Microsoft PowerPoint Edit View Window Help
scGPT talk for webinar — Saved to my Mac
Haotian Cui
scGPT: towards building a foundation model for single-cell multi-omics using generative AI
Authors: Haotian Cui*, Chloe Wang*, Hassaan Maan, Kuan Pang, Fengning Luo, Bo Wang
Present by: Haotian Cui
UNIVERSITY OF TORONTO
VECTOR INSTITUTE
UHN
Thank you for the introduction! I am so thrilled to have the opportunity to present our work scGPT here today. I will start with the introduction to the work and then let's try to see some application demo around zero-shot reference mapping. Hopefully these can give you some idea about how to easily use the model
So let me start and feel free to interrupt with any questions
So scGPT is about building foundation models for single-cell omics

scGPT: towards building a foundation model for single-cell multi-omics using generative AI

Authors: Haotian Cui*, Chloe Wang*, Hassaan Maan, Kuan Pang, Fengning Luo, Bo Wang

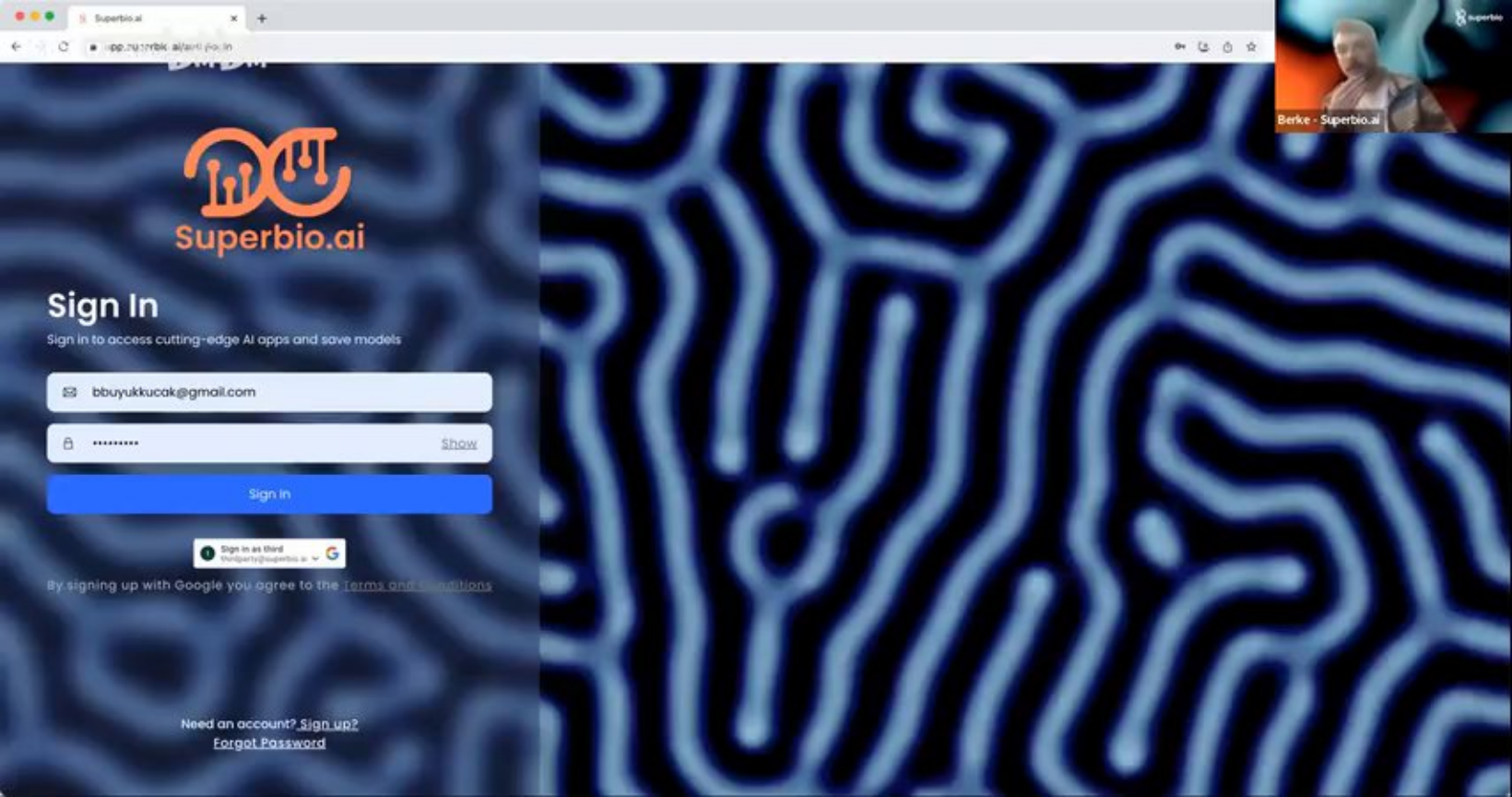
Present by: Haotian Cui

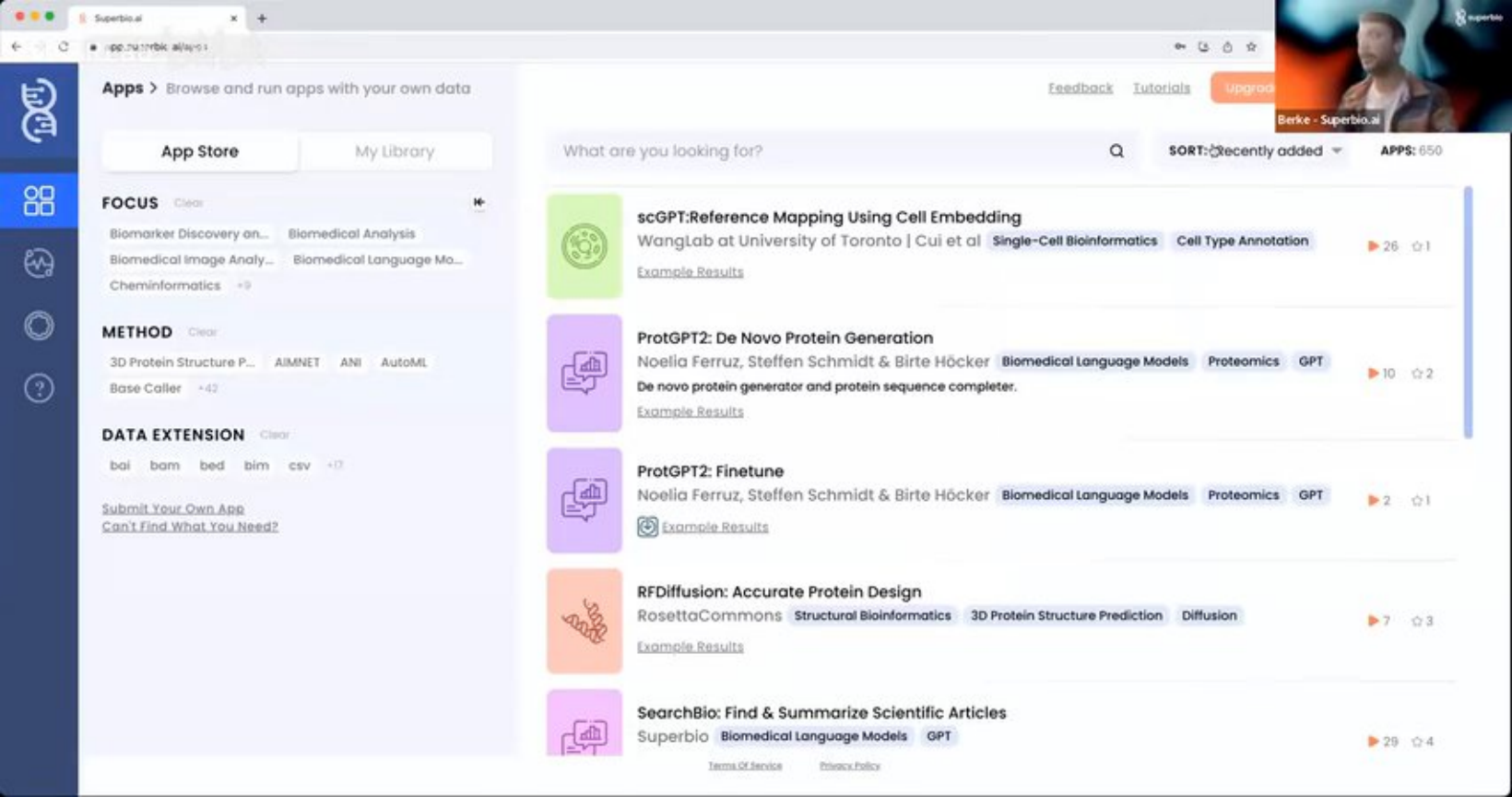
UNIVERSITY OF TORONTO

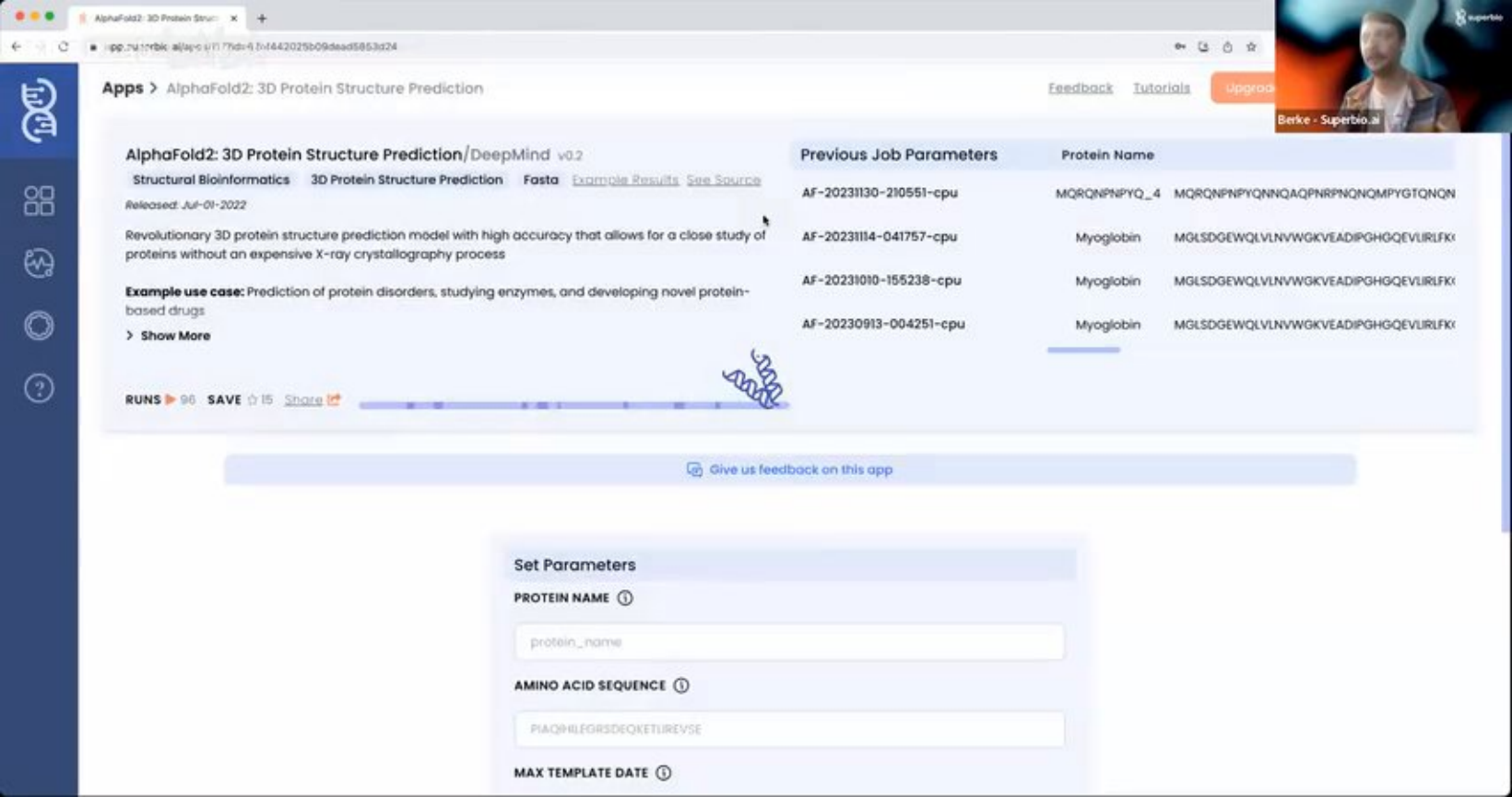
VECTOR INSTITUTE

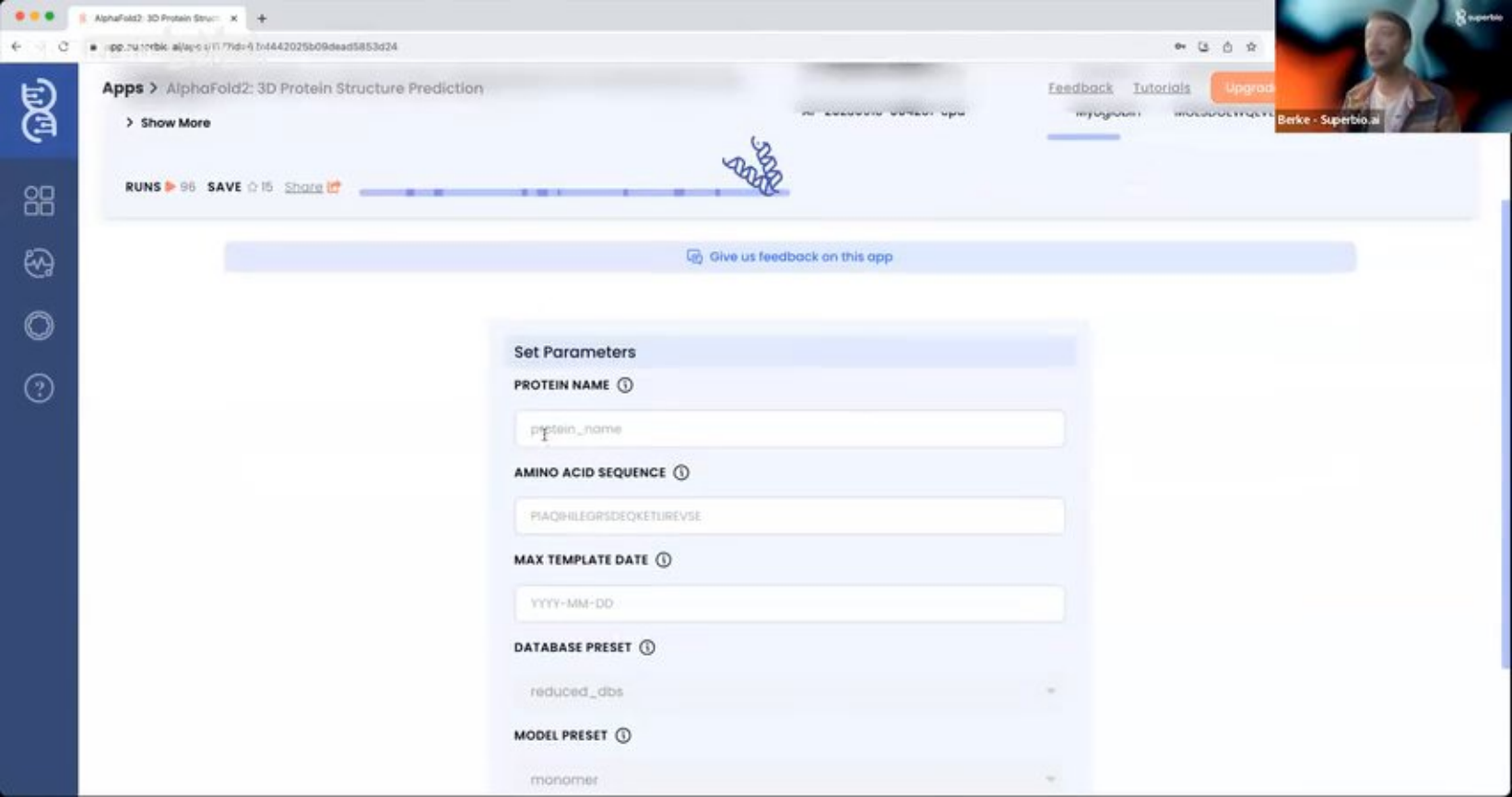
UHN

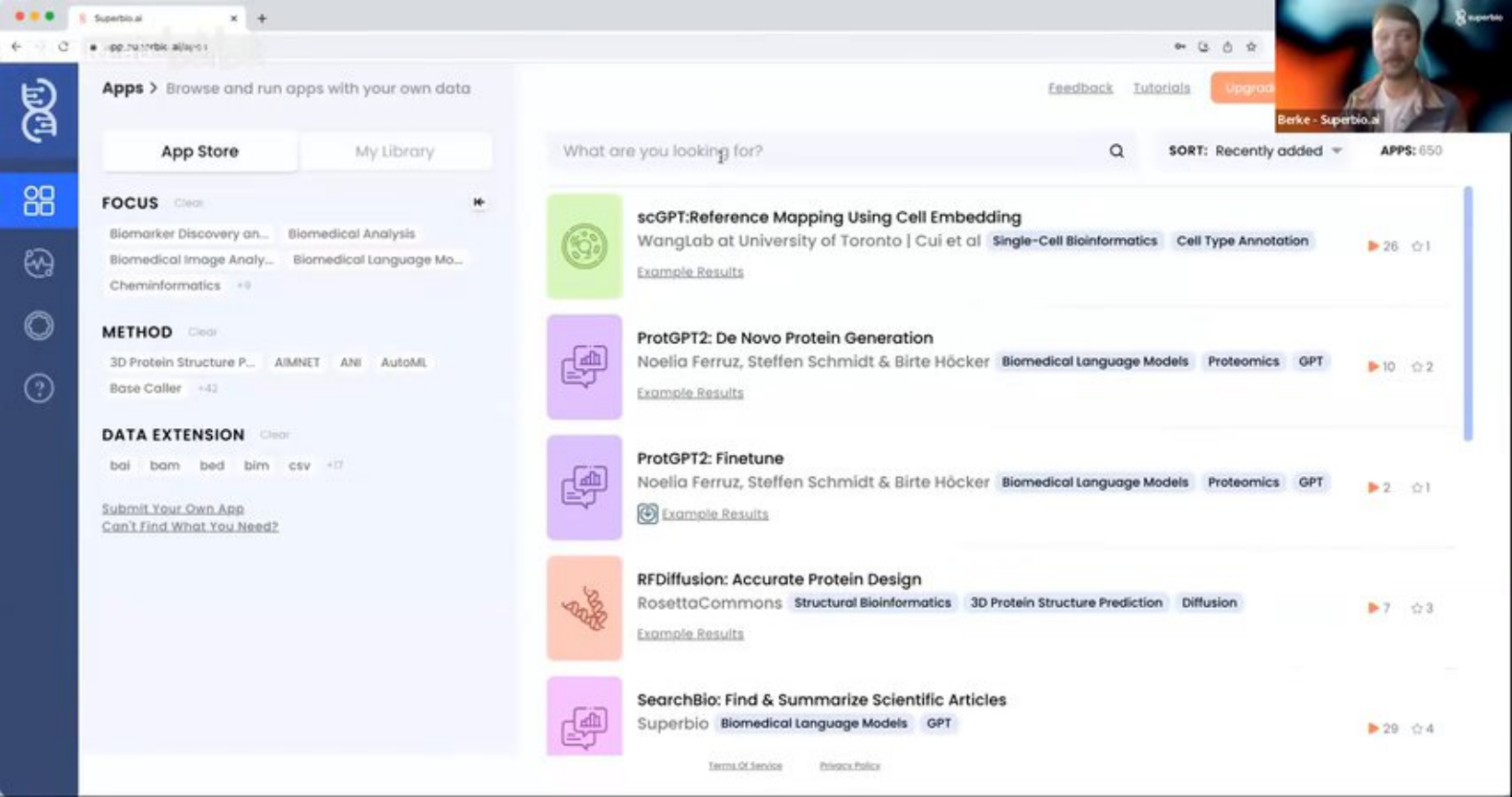

MCBRLab
bilibili
superbio
Berke - Superbio.ai


Superbio.ai
Sign In
Sign in to access cutting-edge AI apps and save models
bbuyukkucak@gmail.com
Show
Sign in
By signing up with Google you agree to the
Need an account? Sign up?
Forgot Password
Berke - Superbio.ai


Apps > Browse and run apps with your own data
Feedback
Tutorials
Berke - Superbio.ai
App Store
My Library
What are you looking for?
SORT: Recently added
APPS: 650
FOCUS
Clear
Biomarker Discovery an...
Biomedical Analysis
Biomedical Image Analy...
Biomedical Language Mo...
Cheminformatics
+9
METHOD
Clear
3D Protein Structure P...
AIMNET
ANI
AutoML
Base Caller
+42
DATA EXTENSION
Clear
bai
bam
bed
bim
csv
+17
Submit Your Own App
Can't Find What You Need?
scGPT:Reference Mapping Using Cell Embedding
WangLab at University of Toronto | Cui et al
Single-Cell Bioinformatics
Cell Type Annotation
26
1
Example Results
ProtGPT2: De Novo Protein Generation
Noelia Ferruz, Steffen Schmidt & Birte Höcker
Biomedical Language Models
Proteomics
GPT
10
2
De novo protein generator and protein sequence completer.
Example Results
ProtGPT2: Finetune
Noelia Ferruz, Steffen Schmidt & Birte Höcker
Biomedical Language Models
Proteomics
GPT
2
1
Example Results
RFDiffusion: Accurate Protein Design
RosettaCommons
Structural Bioinformatics
3D Protein Structure Prediction
Diffusion
7
3
Example Results
SearchBio: Find & Summarize Scientific Articles
Superbio
Biomedical Language Models
GPT
29
4
Terms Of Service
Privacy Policy


Apps > AlphaFold2: 3D Protein Structure Prediction
Feedback
Tutorials
AlphaFold2: 3D Protein Structure Prediction/DeepMind v0.2
Structural Bioinformatics
3D Protein Structure Prediction
Fasta
Example Results
See Source
Released: Jul-01-2022
Revolutionary 3D protein structure prediction model with high accuracy that allows for a close study of proteins without an expensive X-ray crystallography process
Example use case: Prediction of protein disorders, studying enzymes, and developing novel protein-based drugs
Show More
RUNS 96
SAVE 15
Share
Previous Job Parameters
Protein Name
AF-20231130-210551-cpu
MQRQNPNPYQ_4
MQRQNPNPYQNNQAQPNRPNQNQMPYGTQNQN
AF-20231114-041757-cpu
Myoglobin
MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFK
AF-20231010-155238-cpu
Myoglobin
MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFK
AF-20230913-004251-cpu
Myoglobin
MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFK
Give us feedback on this app
Set Parameters
PROTEIN NAME
protein_name
AMINO ACID SEQUENCE
PIAQIHILEGRSDEQKETLIREVSE
MAX TEMPLATE DATE
Berke - Superbio.ai


Apps > AlphaFold2: 3D Protein Structure Prediction
Feedback
Tutorials
Upgrad
Berke - Superbio.ai
Show More
RUNS 96 SAVE 15 Share
Give us feedback on this app
Set Parameters
PROTEIN NAME
protein_name
AMINO ACID SEQUENCE
PIAQIHILEGRSDEQKETLIREVSE
MAX TEMPLATE DATE
YYYY-MM-DD
DATABASE PRESET
reduced_dbs
MODEL PRESET
monomer


Apps > Browse and run apps with your own data
Feedback
Tutorials
Berke - Superbio.ai
App Store
My Library
What are you looking for?
SORT: Recently added
APPS: 650
FOCUS
Biomarker Discovery an...
Biomedical Analysis
Biomedical Image Analy...
Biomedical Language Mo...
Cheminformatics
METHOD
3D Protein Structure P...
AIMNET
ANI
AutoML
Base Caller
DATA EXTENSION
bai
bam
bed
bim
csv
Submit Your Own App
Can't Find What You Need?
scGPT:Reference Mapping Using Cell Embedding
WangLab at University of Toronto | Cui et al
Single-Cell Bioinformatics
Cell Type Annotation
Example Results
ProtGPT2: De Novo Protein Generation
Noelia Ferruz, Steffen Schmidt & Birte Höcker
Biomedical Language Models
Proteomics
GPT
Example Results
ProtGPT2: Finetune
Noelia Ferruz, Steffen Schmidt & Birte Höcker
Biomedical Language Models
Proteomics
GPT
Example Results
RFDiffusion: Accurate Protein Design
RosettaCommons
Structural Bioinformatics
3D Protein Structure Prediction
Diffusion
Example Results
SearchBio: Find & Summarize Scientific Articles
Superbio
Biomedical Language Models
GPT
Terms Of Service
Privacy Policy


MCBRLab
bilibili
Haotian Cui


MCBRLab
bilibili
Haotian Cui


MCBRLab
bilibili
Haotian Cui

scGPT: towards building a foundation model for single-cell multi-omics using generative AI

Authors: Haotian Cui*, Chloe Wang*, Hassaan Maan, Kuan Pang, Fengning Luo, Bo Wang

Present by: Haotian Cui

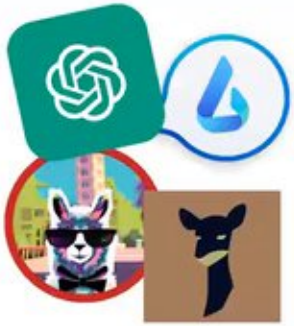
UNIVERSITY OF TORONTO

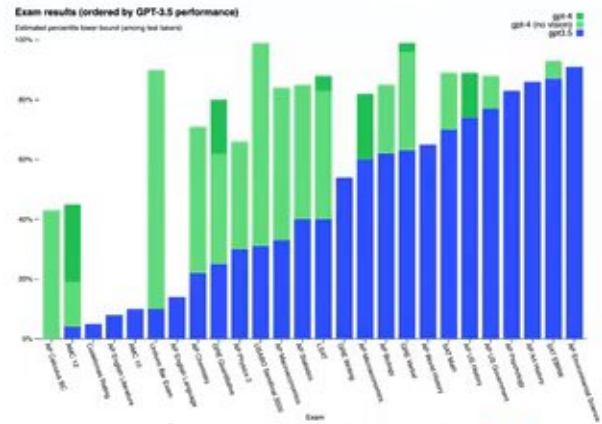
VECTOR INSTITUTE

UHN

Background of foundation models

- Generative pretraining has achieved remarkable success in various domains such as natural language processing and computer vision
- Specifically, the combination of large-scale diverse datasets and pre-trained transformers has emerged as a promising approach for developing foundation models.

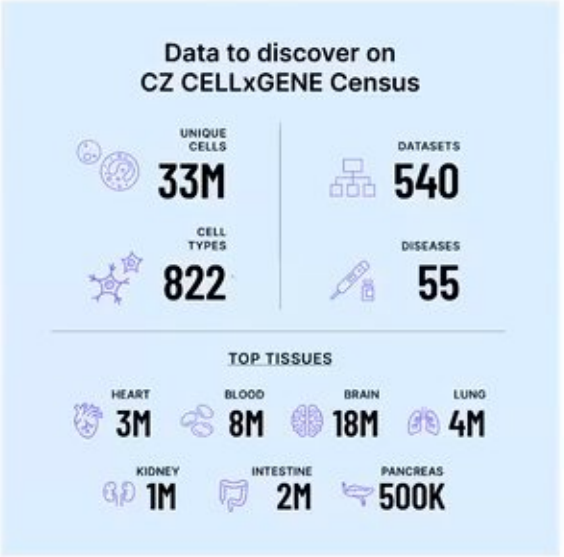




GPT-4 performances on standard exams. Credit to OpenAI

Potential of foundation models for single-cell biol

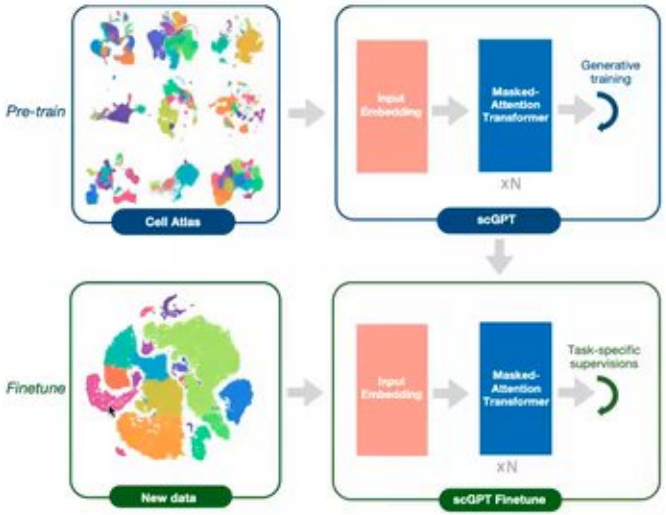
- While texts are made up of words, cells can be characterized by genes.
- This analogy inspires us to explore foundation models for single cell. In the long run, we expect such model has the potential to gain general understanding of molecular biology.
- This is feasible based on two considerations:
  - Transformers would a suitable to learn interactions between gene tokens.
  - The vast scale of sequencing data, of diverse cell types and conditions, and it is growing exponentially.



We are particularly excited about the recent launch of CellxGene census. Figure credit to @JCoolScience

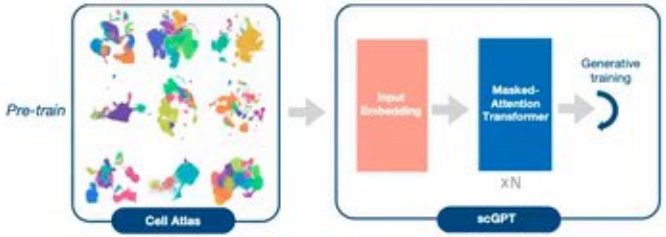
Overview of scGPT

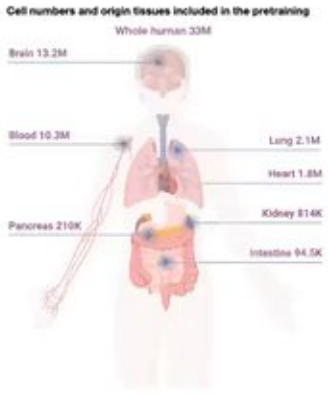
- We present scGPT to construct a single-cell foundation model. The workflow contains two steps:
  1. Generative **pre-training** on over 33 million normal human cells
  2. **Fine-tuning** for downstream tasks



Generative pretraining in scGPT

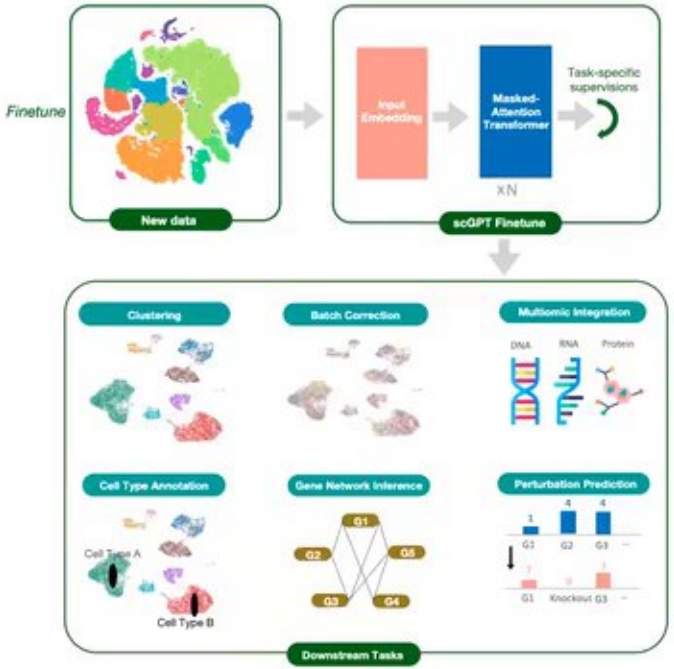
- In the pre-training:
  - Large-scale expression matrices were collected and preprocessed
  - We selected around 33 million normal human cells of 51 organs/tissues from the CZ CellxGene collection
  - Numbers of cells from major tissues shown on the right





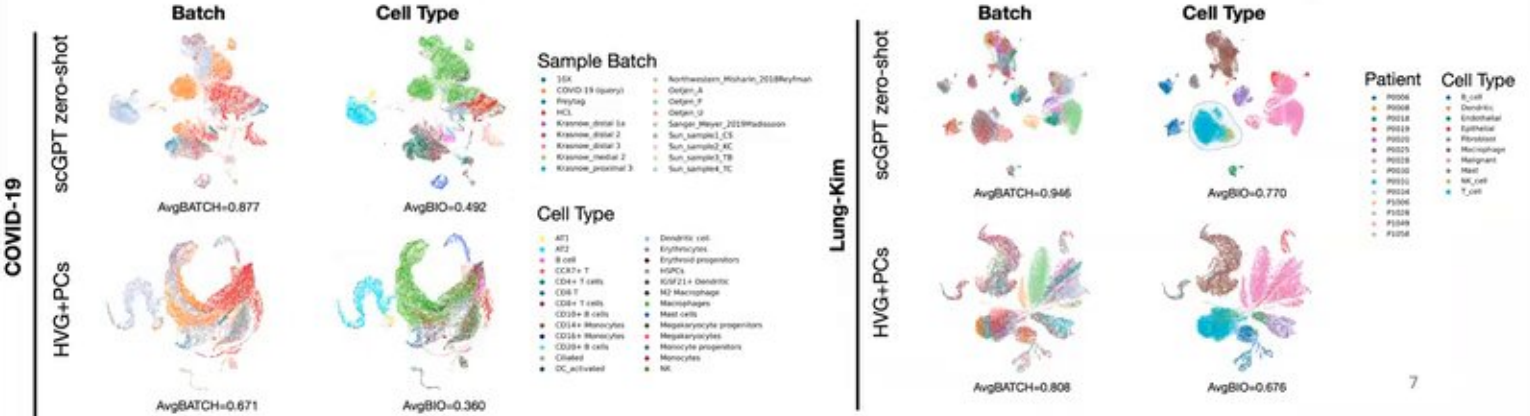
Overview of fine-tuning in scGPT

- In fine-tuning:
  - A set of versatile learning objectives are adopted to readily support various applications
  - Self-supervised objectives:
    - Gene Expression Prediction (GEP)
    - Gene Expression Prediction for Cell Modelling (GEPC)
    - Elastic Cell Similarity (ECS)
  - Supervised objectives:
    - Domain Adaptation via Reverse Back-propagation (DAR)
    - Cell Type Classification (CLS)



Zero-shot applications

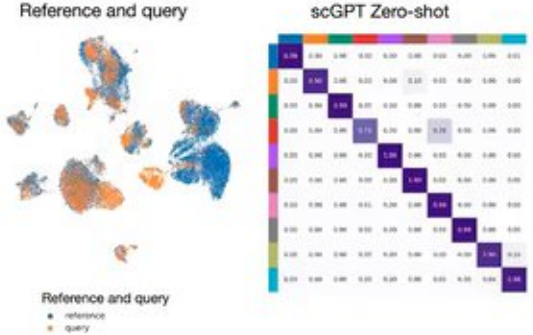
- **Zero-shot**, meaning using the pretrained model to generate embeddings for new data, without any further training, **much faster, more accessible**
- UMAP of zero-shot embeddings on two disease datasets, compared to HVGs
  - Considerable ability to distinguish cell types
  - Note the pretraining process is not designed for mitigating technical batch effects



Zero-shot reference mapping

- We also use the embeddings in a reference mapping manner
- We split the data to simulate a setting that some of the patient cells, query set, were not annotated. Cell types were propagated from reference data set to label the query set.
- The zero-shot workflow demonstrates competitive accuracy across comparisons.

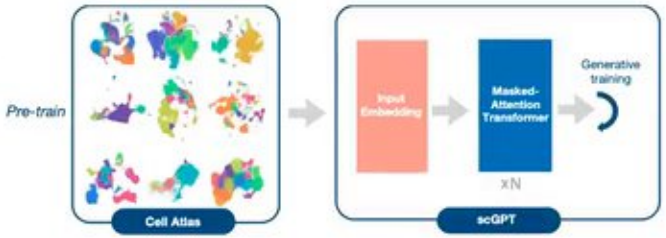
Lung-Kim

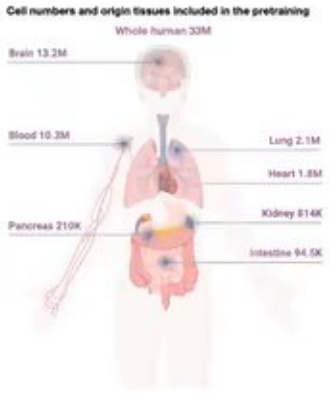


Dataset	Method	Evaluation Metrics			
		Accuracy	*Precision*	*Recall*	*MacroF1*
COVID-19	scGPT (fine-tuned)	**0.891**	**0.577**	**0.567**	**0.547**
	scGPT (zero-shot)	0.867	0.513	0.476	0.468
	Azimuth	0.878	0.515	0.418	0.444
	expiMap	0.730	0.441	0.325	0.335
Lung-Kim	scGPT (fine-tuned)	**0.974**	0.959	**0.967**	**0.962**
	scGPT (zero-shot)	0.968	**0.970**	0.933	0.948
	Azimuth	0.970	0.957	0.962	0.959
	expiMap	0.920	0.940	0.846	0.878

Generative pretraining in scGPT

- In the pre-training:
  - Large-scale expression matrices were collected and preprocessed
  - We selected around 33 million normal human cells of 51 organs/tissues from the CZ CellxGene collection
  - Numbers of cells from major tissues shown on the right

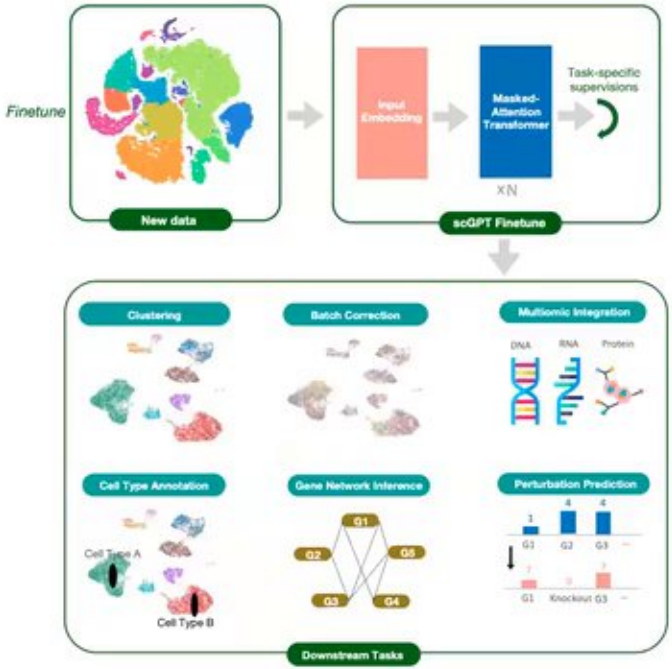




MCBRLab

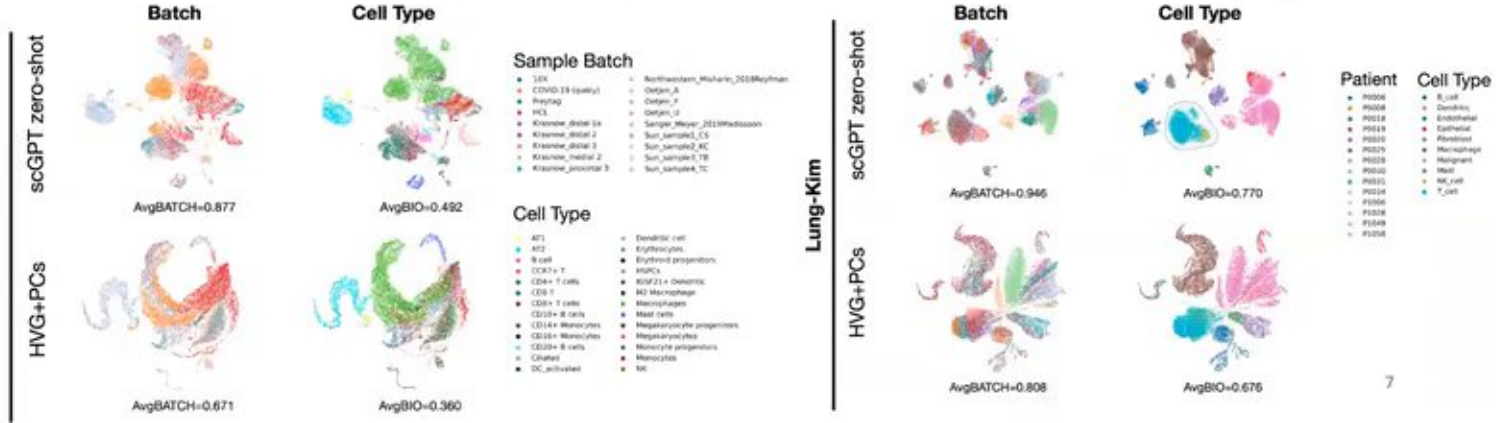
Overview of fine-tuning in scGPT

- In fine-tuning:
  - A set of versatile learning objectives are adopted to readily support various applications
  - Self-supervised objectives:
    - Gene Expression Prediction (GEP)
    - Gene Expression Prediction for Cell Modelling (GEPC)
    - Elastic Cell Similarity (ECS)
  - Supervised objectives:
    - Domain Adaptation via Reverse Back-propagation (DAR)
    - Cell Type Classification (CLS)



Zero-shot applications

- **Zero-shot**, meaning using the pretrained model to generate embeddings for new data, without any further training, **much faster, more accessible**
- UMAP of zero-shot embeddings on two disease datasets, compared to HVGs
  - Considerable ability to distinguish cell types
  - Note the pretraining process is not designed for mitigating technical batch effects



Zero-shot reference mapping

- We also use the embeddings in a reference mapping manner
- We split the data to simulate a setting that some of the patient cells, query set, were not annotated. Cell types were propagated from reference data set to label the query set.
- The zero-shot workflow demonstrates competitive accuracy across comparisons.

Lung-Kim

Dataset	Method	Evaluation Metrics			
		Accuracy	*Precision*	*Recall*	*MacroF1*
COVID-19	scGPT (fine-tuned)	**0.891**	**0.577**	**0.567**	**0.547**
	scGPT (zero-shot)	0.867	0.513	0.476	0.468
	Azimuth	0.878	0.515	0.418	0.444
	expiMap	0.730	0.441	0.325	0.335
Lung-Kim	scGPT (fine-tuned)	**0.974**	0.959	**0.967**	**0.962**
	scGPT (zero-shot)	0.968	**0.970**	0.933	0.948
	Azimuth	0.970	0.957	0.962	0.959
	expiMap	0.920	0.940	0.846	0.878

Zero-shot ref map to cell atlas

- Zero-shot embeddings can further be used to map cells and propagate metainfo from large cell atlas
- For this new feature, we generated the cell embeddings for millions of cells on CellXGene, and build an index for efficient similarity search using the faiss toolbox.
- This can be used to query new cells super fast, https://github.com/bowang-lab/scGPT/blob/main/tutorials/Tutorial_Reference_Mapping.ipynb

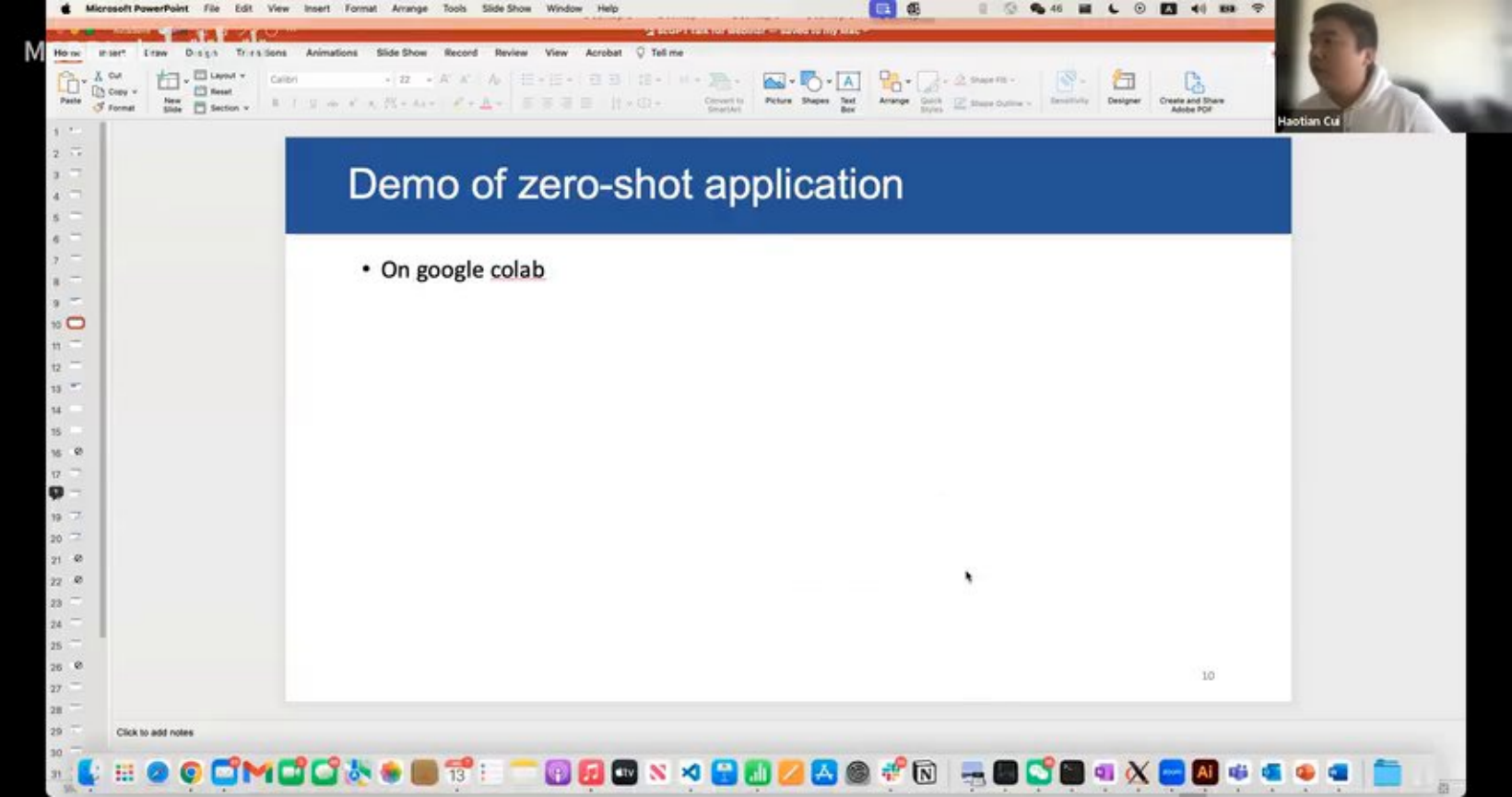
The search runs super fast, especially on GPU. Here the similarity search for 4,000 query cells within the whole reference of millions should take around 7 second on CPU and 0.1 second on GPU.

In [10]:
```
%%time
k = 50
# test with the first 100 cells
distances, idx = index.search(test_emebd, k)
```

```
CPU times: user 252 ms, sys: 493 ms, total: 745 ms
Wall time: 133 ms
```

Demo of zero-shot application

- On google colab


Microsoft PowerPoint File Edit View Insert Format Arrange Tools Slide Show Window Help
Home Insert Draw Design Transitions Animations Slide Show Record Review View Acrobat Tell me
Haotian Cui
Demo of zero-shot application
• On google colab
10
Click to add notes

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

Zero-shot reference mapping tutorial with scGPT

Introduction

This tutorial covers the zero-shot reference mapping with scGPT. This workflow achieves accurate and fast reference mapping for scRNA-seq datasets without fine-tuning (or any extensive training) of scGPT. To further boost the performance, we recommend fine-tuning scGPT.

We will use COVID-19 dataset to demonstrate the zero-shot reference mapping. You can download the processed reference and query datasets from here. The COVID-19 dataset is derived from the work by Lotfollahi et al, which contains 18 distinct batches and diverse samples from lung tissues. The reference dataset consists of 15,997 cells and the query dataset contains 4,003 cells. You may place the dataset under `data` directory in the outer level.

Particularly, we use the `scGPT_human` model to provide embeddings out of the box. You may download it from here.

The zero-shot reference mapping workflow is as follows:

1. Load and pre-process the dataset
2. Generate scGPT embeddings for each cell in reference and query datasets
3. Transfer the annotations from reference to query dataset

At the appendix of this tutorial, we will also showcase the zero-shot reference mapping on Lung dataset. You can find the dataset here.

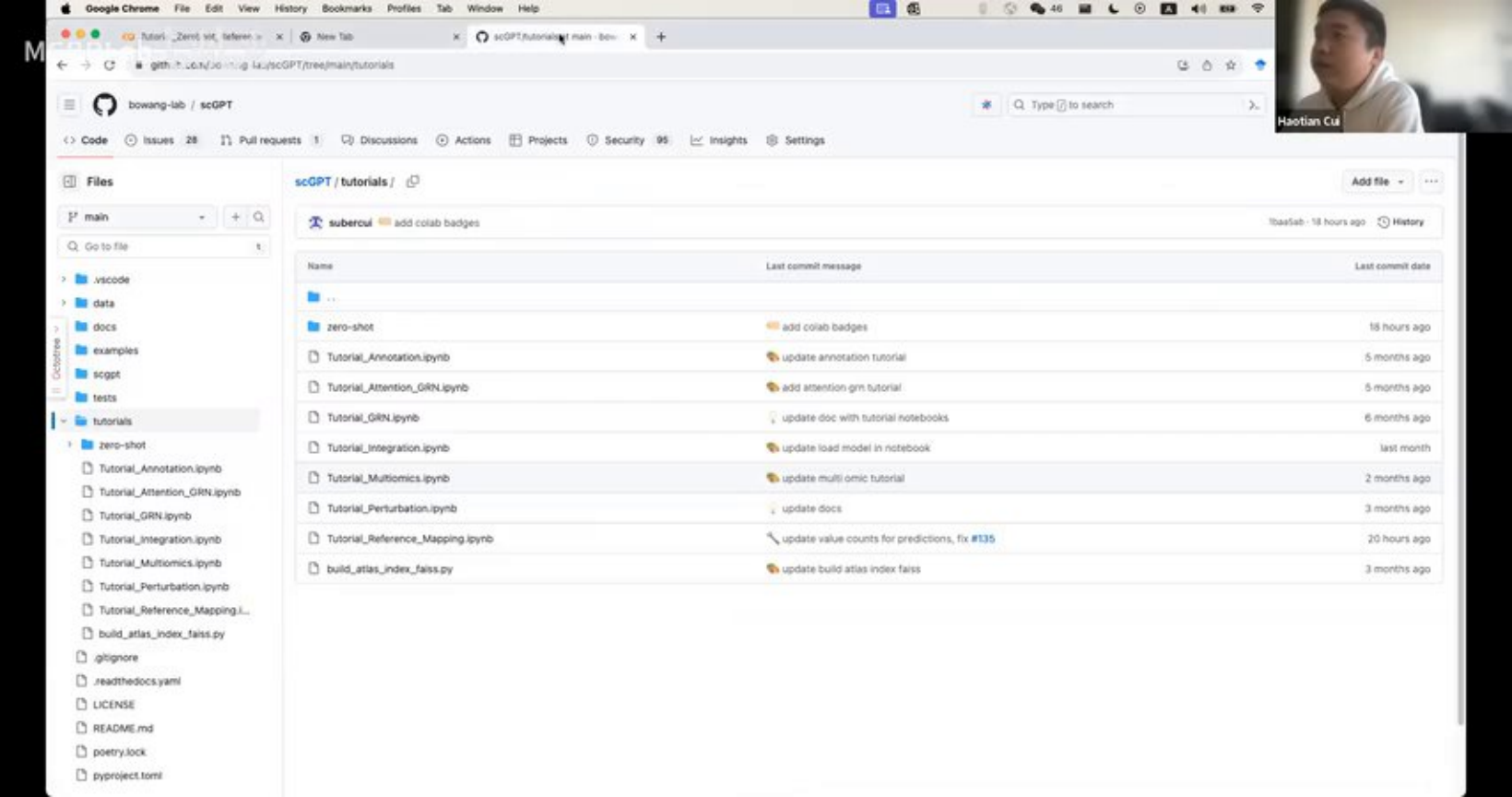
We use a similarity-based method for transferring the annotation, which involves comparing the similarity between the query cell embedding and the reference cell embeddings. We use FAISS to perform the similarity search.

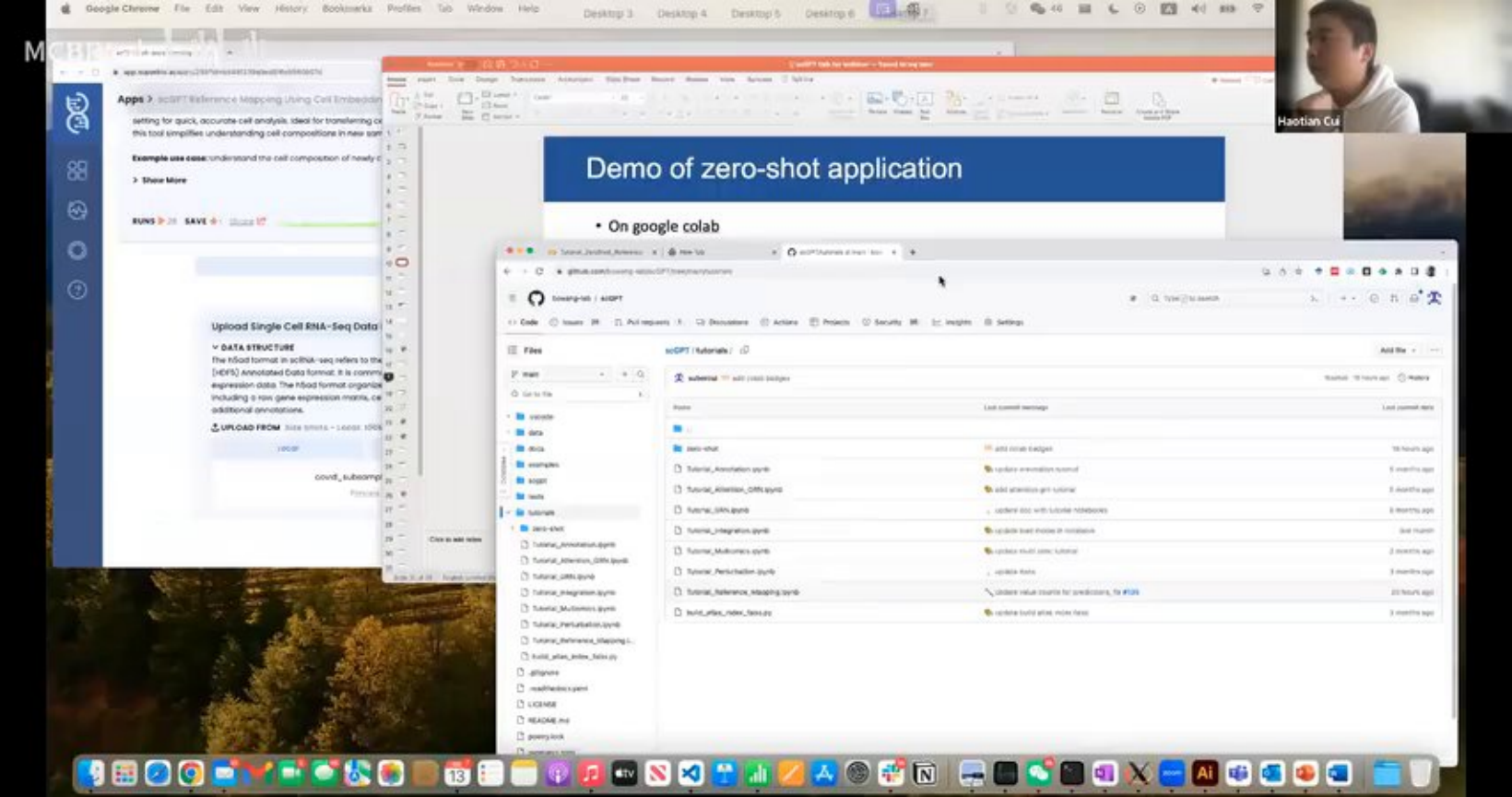
Open in Colab

[1] # Specifically for Google Colab, install dependencies and download data

```
import os
import sys

if "google.colab" in sys.modules:
    print("Running on Google Colab")
    print("Installing dependencies...")
    !pip install -U scgpt
```

0s completed at 12:03 PM


Haotian Cui
bowang-lab / scGPT
Code
Issues 28
Pull requests 1
Discussions
Actions
Projects
Security 95
Insights
Settings
Files
main
scGPT / tutorials /
Add file
subercui add colab badges
Name
Last commit message
Last commit date
zero-shot
add colab badges
18 hours ago
Tutorial_Annotation.ipynb
update annotation tutorial
5 months ago
Tutorial_Attention_GRN.ipynb
add attention grn tutorial
5 months ago
Tutorial_GRN.ipynb
update doc with tutorial notebooks
6 months ago
Tutorial_Integration.ipynb
update load model in notebook
last month
Tutorial_Multiomics.ipynb
update multi omic tutorial
2 months ago
Tutorial_Perturbation.ipynb
update docs
3 months ago
Tutorial_Reference_Mapping.ipynb
update value counts for predictions, fix #135
20 hours ago
build_atlas_index_faiss.py
update build atlas index faiss
3 months ago


Haotian Cui
Demo of zero-shot application
• On google colab

Zero-shot reference mapping

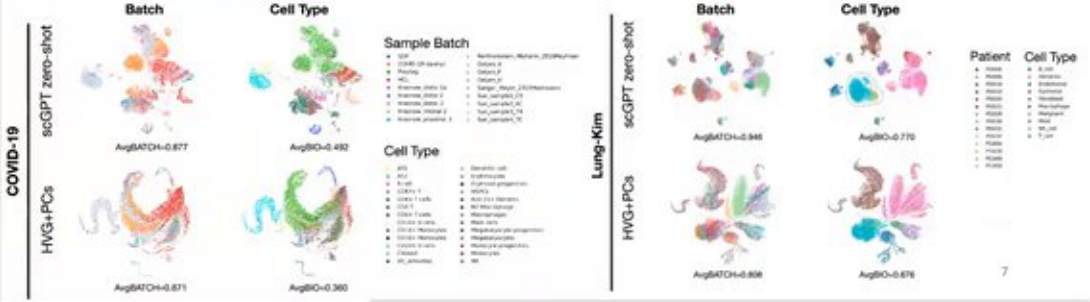
- We also use the embeddings in a reference mapping manner
- We split the data to simulate a setting that some of the patient cells, query set, were not annotated. Cell types were propagated from reference data set to label the query set.
- The zero-shot workflow demonstrates competitive accuracy across comparisons.

Dataset	Method	Evaluation Metrics *Accuracy*	*Precision*	*Recall*	*MacroF1*
COVID-19	scGPT (fine-tuned)	**0.891**	**0.577**	**0.567**	**0.547**
	scGPT (zero-shot)	0.867	0.513	0.476	0.468
	Azimuth	0.878	0.515	0.418	0.444
	expiMap	0.730	0.441	0.325	0.335
Lung-Kim	scGPT (fine-tuned)	**0.974**	0.959	**0.967**	**0.962**
	scGPT (zero-shot)	0.968	**0.970**	0.933	0.948
	Azimuth	0.970	0.957	0.962	0.959
	expiMap	0.920	0.940	0.846	0.878

Click to add notes

Zero-shot applications

- **Zero-shot**, meaning using the pretrained model to generate embeddings for new data, without any further training, **much faster, more accessible**
- UMAP of zero-shot embeddings on two disease datasets, compared to HVGs
  - Considerable ability to distinguish cell types
  - Note the pretraining process is not designed for mitigating technical batch effects

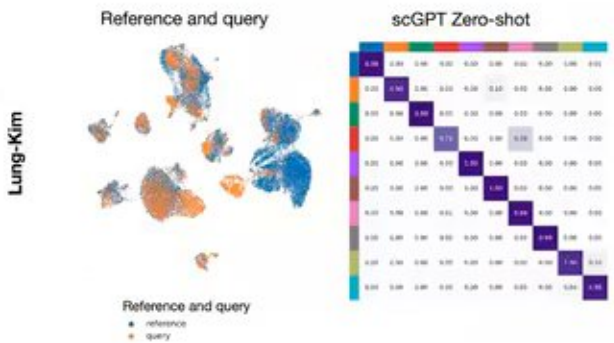


Have been working new extensions exploring the possibility of zero-shot applications. Here, I'd like to share some new results

Particularly want to mention the influence of pretraining,
so of the technical info is actually signals

Zero-shot reference mapping

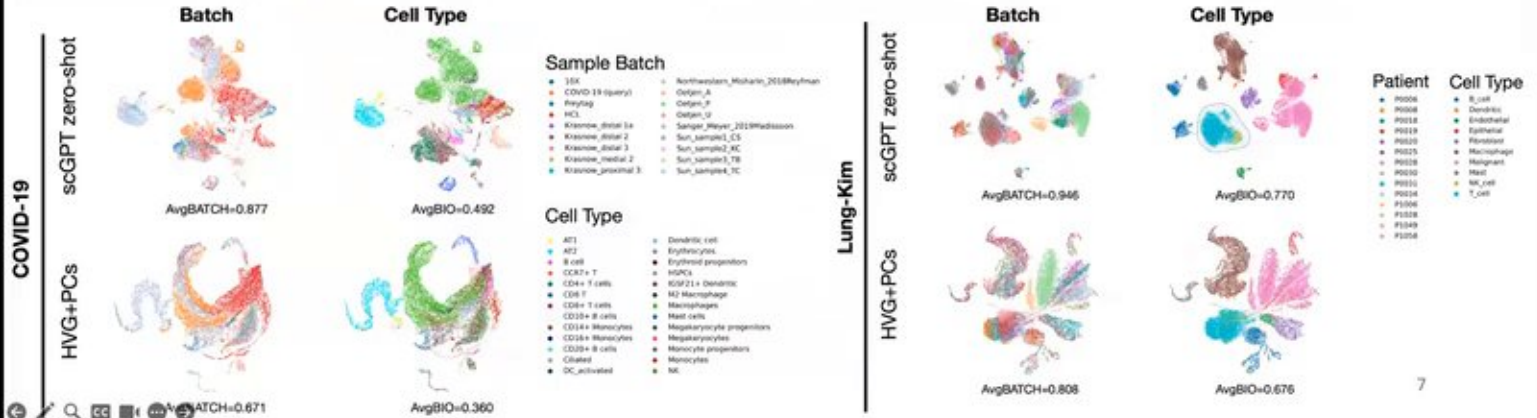
- We also use the embeddings in a reference mapping manner
- We split the data to simulate a setting that some of the patient cells, query set, were not annotated. Cell types were propagated from reference data set to label the query set.
- The zero-shot workflow demonstrates competitive accuracy across comparisons.



Dataset	Method	Evaluation Metrics			
		Accuracy	*Precision*	*Recall*	*MacroF1*
COVID-19	scGPT (fine-tuned)	**0.891**	**0.577**	**0.567**	**0.547**
	scGPT (zero-shot)	0.867	0.513	0.476	0.468
	Azimuth	0.878	0.515	0.418	0.444
	expiMap	0.730	0.441	0.325	0.335
Lung-Kim	scGPT (fine-tuned)	**0.974**	0.959	**0.967**	**0.962**
	scGPT (zero-shot)	0.968	**0.970**	0.933	0.948
	Azimuth	0.970	0.957	0.962	0.959
	expiMap	0.920	0.940	0.846	0.878

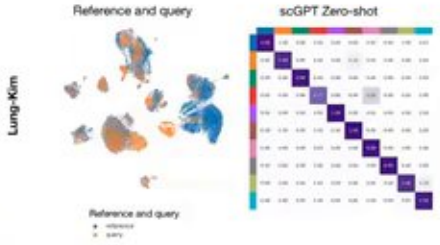
Zero-shot applications

- **Zero-shot**, meaning using the pretrained model to generate embeddings for new data, without any further training, **much faster, more accessible**
- UMAP of zero-shot embeddings on two disease datasets, compared to HVGs
  - Considerable ability to distinguish cell types
  - Note the pretraining process is not designed for mitigating technical batch effects



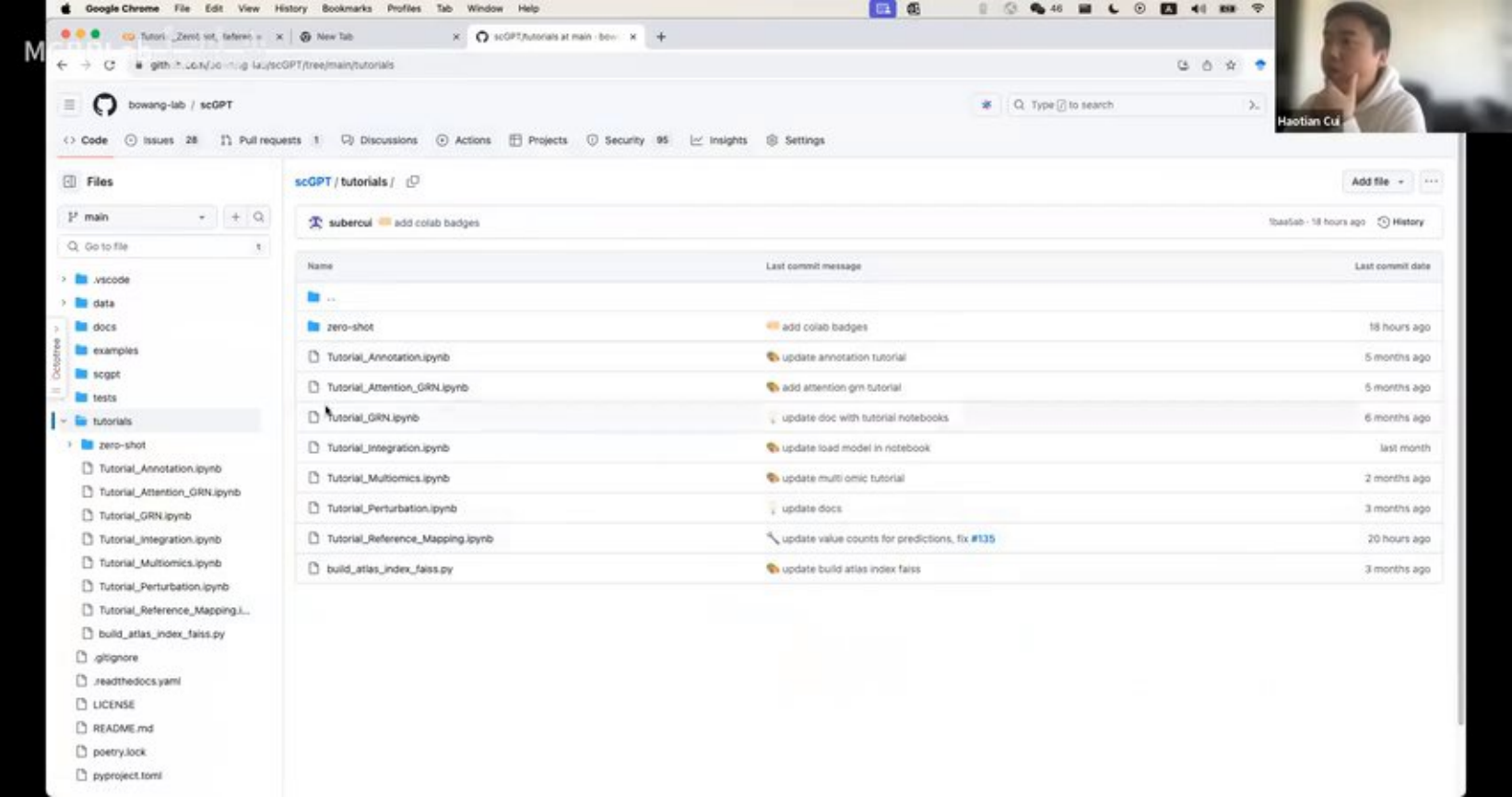
Zero-shot reference mapping

- We also use the embeddings in a reference mapping manner
- We split the data to simulate a setting that some of the patient cells, query set, were not annotated. Cell types were propagated from reference data set to label the query set.
- The zero-shot workflow demonstrates competitive accuracy across comparisons.



Dataset	Method	Evaluation Metrics			
		Accuracy	*Precision*	*Recall*	*MacroF1*
COVID-19	scGPT (fine-tuned)	**0.891**	**0.577**	**0.567**	**0.547**
	scGPT (zero-shot)	0.867	0.513	0.476	0.468
	Azimuth	0.878	0.515	0.418	0.444
	expiMap	0.730	0.441	0.325	0.335
Lung-Kim	scGPT (fine-tuned)	**0.974**	0.959	**0.967**	**0.962**
	scGPT (zero-shot)	0.968	**0.970**	0.933	0.948
	Azimuth	0.970	0.957	0.962	0.959
	expiMap	0.920	0.940	0.846	0.878

Click to add notes


Haotian Cui
bowang-lab / scGPT
Code
Issues 28
Pull requests 1
Discussions
Actions
Projects
Security 95
Insights
Settings
Files
main
scGPT / tutorials /
Add file
subercui add colab badges
Name
Last commit message
Last commit date
zero-shot
add colab badges
18 hours ago
Tutorial_Annotation.ipynb
update annotation tutorial
5 months ago
Tutorial_Attention_GRN.ipynb
add attention grn tutorial
5 months ago
Tutorial_GRN.ipynb
update doc with tutorial notebooks
6 months ago
Tutorial_Integration.ipynb
update load model in notebook
last month
Tutorial_Multiomics.ipynb
update multi omic tutorial
2 months ago
Tutorial_Perturbation.ipynb
update docs
3 months ago
Tutorial_Reference_Mapping.ipynb
update value counts for predictions, fix #135
20 hours ago
build_atlas_index_faiss.py
update build atlas index faiss
3 months ago
.vscode
data
docs
examples
scgpt
tests
tutorials
.gitignore
.readthedocs.yaml
LICENSE
README.md
poetry.lock
pyproject.toml

Files

main

Go to file

- .vscode
- data
- docs
- examples
- scgpt
- tests
- tutorials
- zero-shot
  - Tutorial_ZeroShot_Integratio...
  - Tutorial_ZeroShot_Reference...
  - Tutorial_Annotation.ipynb
  - Tutorial_Attention_GRN.ipynb
  - Tutorial_GRN.ipynb
  - Tutorial_Integration.ipynb
  - Tutorial_Multiomics.ipynb
  - Tutorial_Perturbation.ipynb
  - Tutorial_Reference_Mapping.i...
  - build_atlas_index_faiss.py
- .gitignore
- .readthedocs.yaml
- LICENSE
- README.md
- poetry.lock
- pyproject.toml

Preview | Code | Blame | 705 lines (705 loc) · 1.42 MB

Zero-shot reference mapping tutorial with scGPT

Introduction

This tutorial covers the zero-shot reference mapping with scGPT. This workflow achieves accurate and fast reference mapping for scRNA-seq datasets without fine-tuning (or any extensive training) of scGPT. To further boost the performance, we recommend fine-tuning scGPT.

We will use COVID-19 dataset to demonstrate the zero-shot reference mapping. You can download the processed reference and query datasets from here. The COVID-19 dataset is derived from the work by Lotfollahi et al, which contains 18 distinct batches and diverse samples from lung tissues. The reference dataset consists of 15,997 cells and the query dataset contains 4,003 cells. You may place the dataset under `data` directory in the outer level.

Particularly, we use the `scGPT_human` model to provide embeddings out of the box. You may download it from here.

The zero-shot reference mapping workflow is as follows:

1. Load and pre-process the dataset
2. Generate scGPT embeddings for each cell in reference and query datasets
3. Transfer the annotations from reference to query dataset

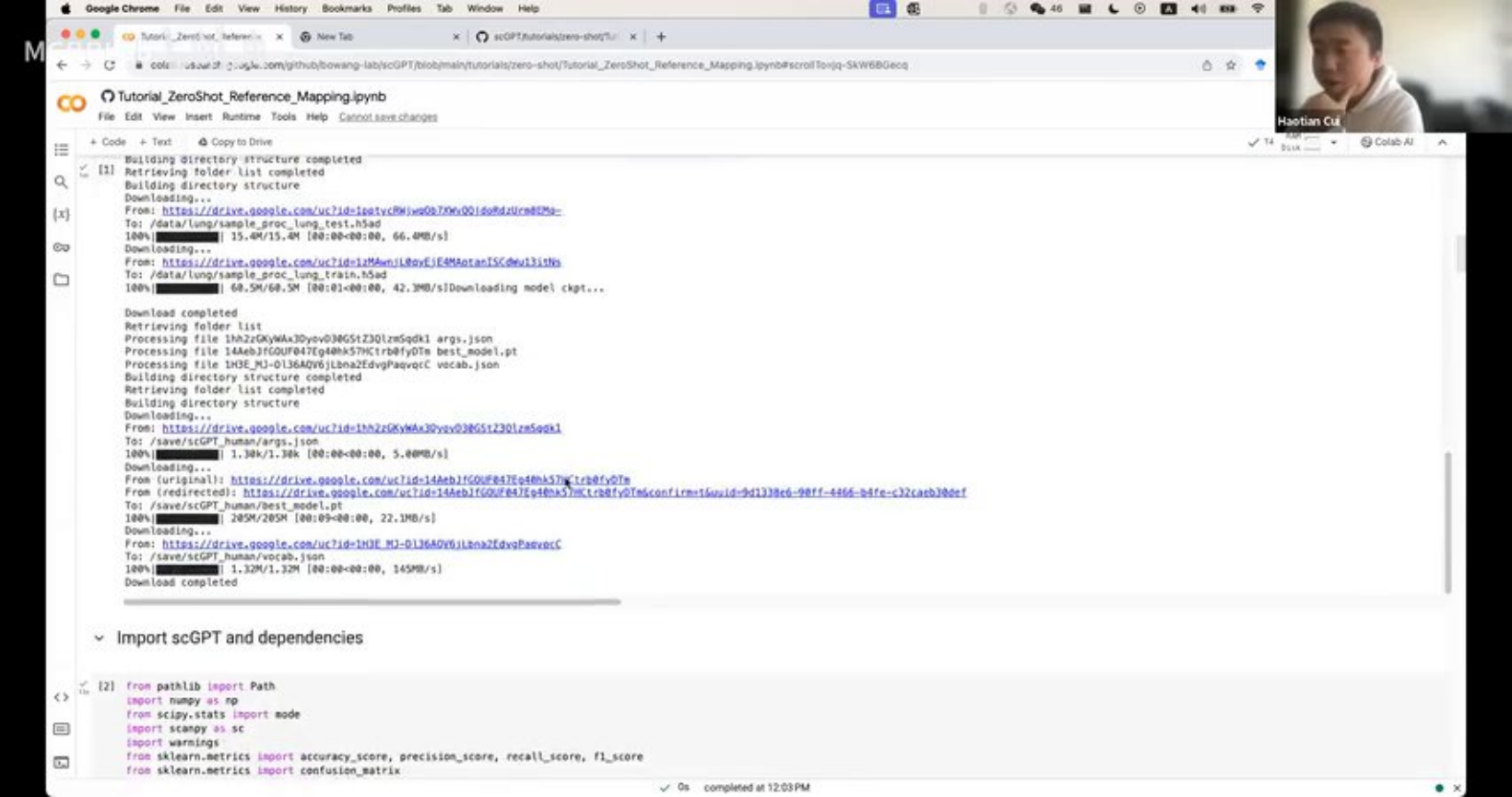
At the appendix of this tutorial, we will also showcase the zero-shot reference mapping on Lung dataset. You can find the dataset here.

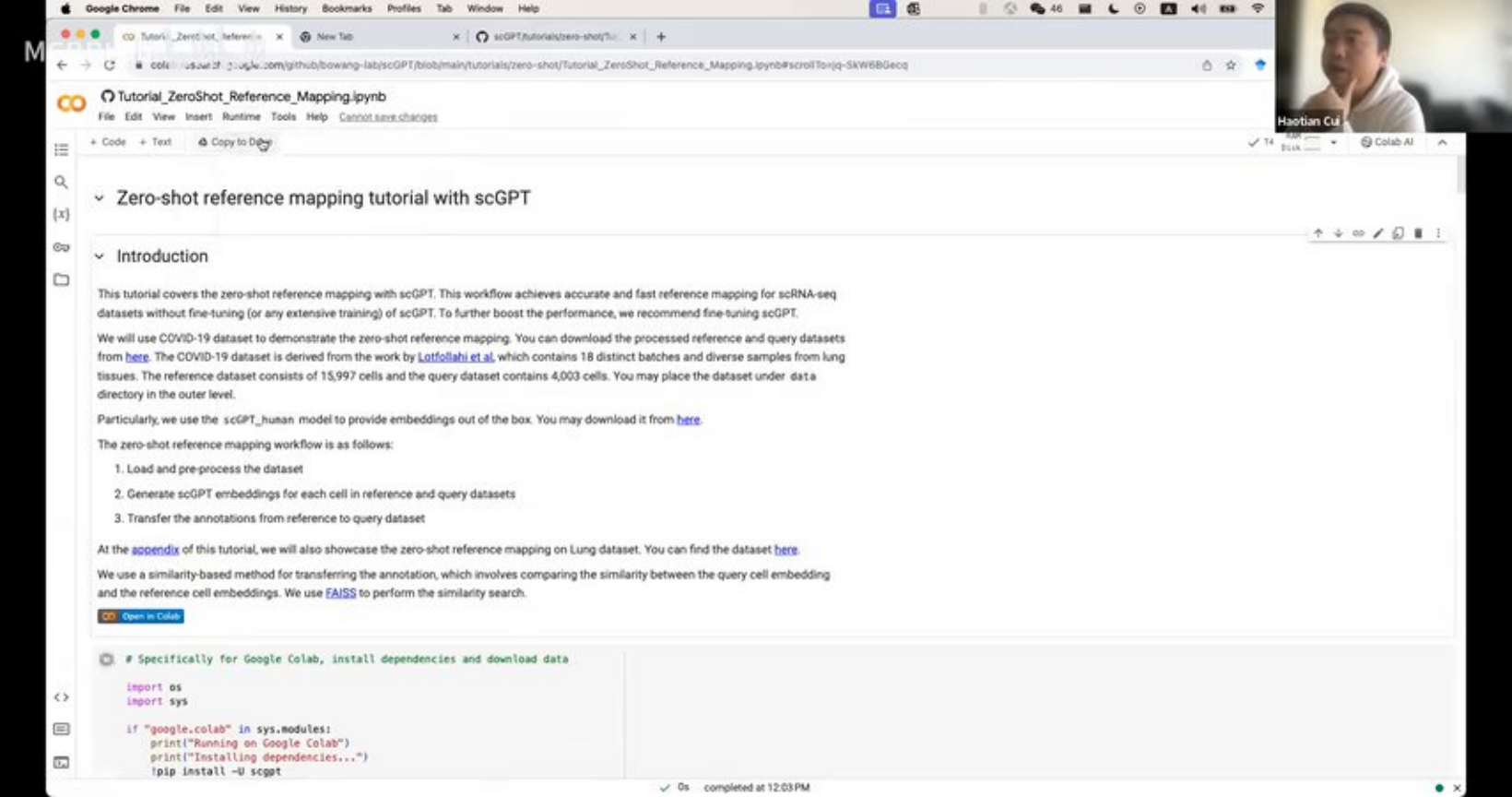
We use a similarity-based method for transferring the annotation, which involves comparing the similarity between the query cell embedding and the reference cell embeddings. We use FAISS to perform the similarity search.

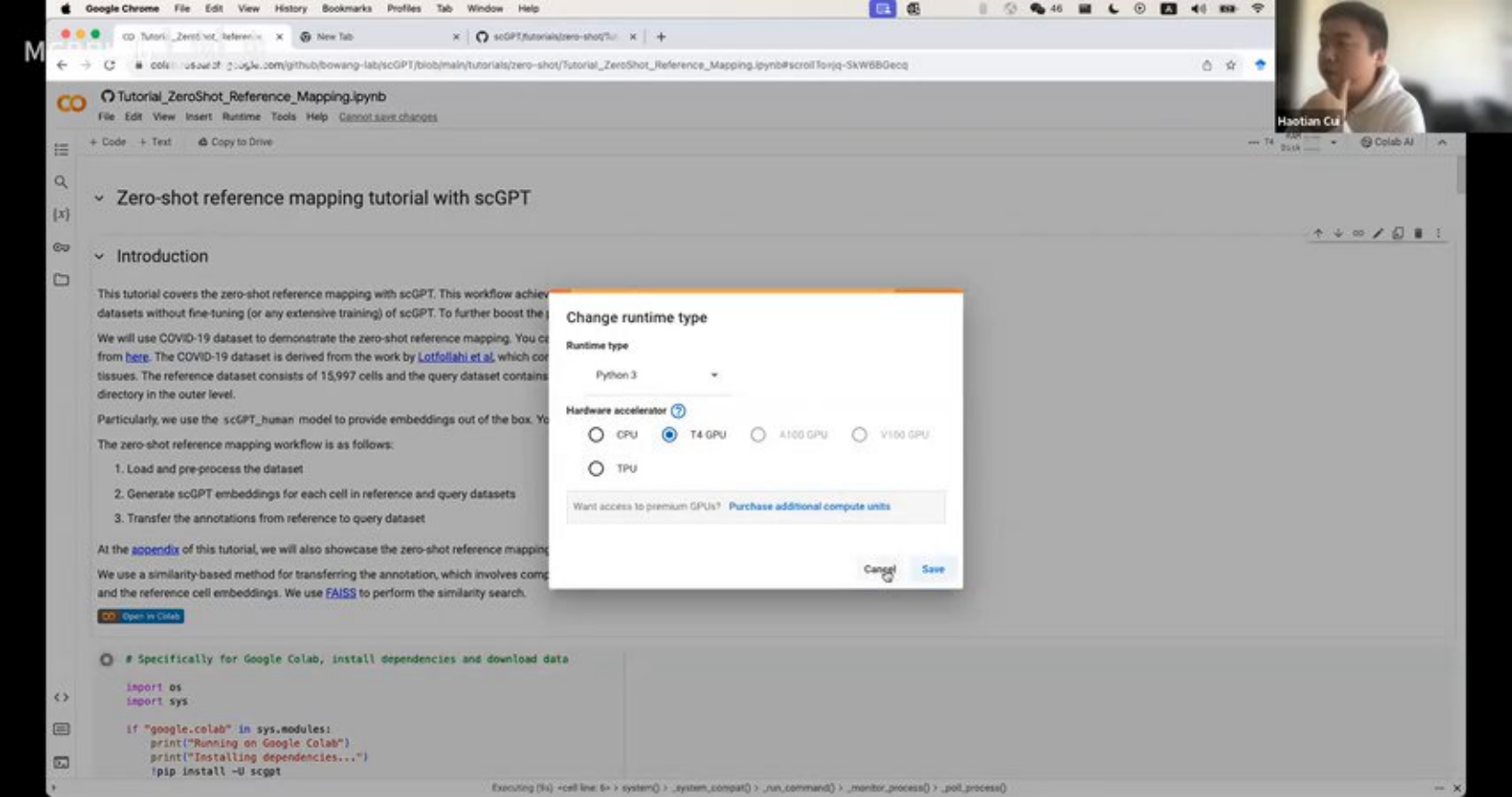
Open in Colab

In [1]:

```
# Specifically for Google Colab, install dependencies and download data

import os
import sys

if "google.colab" in sys.modules:
    print("Running on Google Colab")
    print("Installing dependencies...")
    !pip install -U scgpt
    # the optional dependency of flash-attion is skipped on colab
    !pip install wandb louvain faiss-cpu

    # NOTE: May need to restart runtime after the installation
```


Haotian Cui
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
Building directory structure completed
Retrieving folder list completed
Building directory structure
Downloading...
From: https://drive.google.com/uc?id=1patycRWjwqOb7XWvQQ1doRdzUrm8EMa-
To: /data/lung/sample_proc_lung_test.h5ad
100%| | 15.4M/15.4M [00:00<00:00, 66.4MB/s]
Downloading...
From: https://drive.google.com/uc?id=1zMAwnjL8qvEjE4MAotanISCdWu13itNs
To: /data/lung/sample_proc_lung_train.h5ad
100%| | 60.5M/60.5M [00:01<00:00, 42.3MB/s]Downloading model ckpt...
Download completed
Retrieving folder list
Processing file 1hh2zGKyWAx3OyovO30GStZ3QlzmSqdk1 args.json
Processing file 14AebJfGOUF047Eg40hk57HCtrb0fyDTm best_model.pt
Processing file 1H3E_MJ-Dl36AQV6jLbna2EdvgPaqvqcC vocab.json
Building directory structure completed
Retrieving folder list completed
Building directory structure
Downloading...
From: https://drive.google.com/uc?id=1hh2zGKyWAx3OyovO30GStZ3QlzmSqdk1
To: /save/scGPT_human/args.json
100%| | 1.30k/1.30k [00:00<00:00, 5.00MB/s]
Downloading...
From (original): https://drive.google.com/uc?id=14AebJfGOUF047Eg40hk57HCtrb0fyDTm
From (redirected): https://drive.google.com/uc?id=14AebJfGOUF047Eg40hk57HCtrb0fyDTm&confirm=t&uuid=9d1338e6-90ff-4466-b4fe-c32caeb300ef
To: /save/scGPT_human/best_model.pt
100%| | 205M/205M [00:09<00:00, 22.1MB/s]
Downloading...
From: https://drive.google.com/uc?id=1H3E_MJ-Dl36AQV6jLbna2EdvgPaqvqcC
To: /save/scGPT_human/vocab.json
100%| | 1.32M/1.32M [00:00<00:00, 145MB/s]
Download completed
Import scGPT and dependencies
[2] from pathlib import Path
import numpy as np
from scipy.stats import mode
import scanpy as sc
import warnings
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix
0s completed at 12:03 PM


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
Zero-shot reference mapping tutorial with scGPT
Introduction
This tutorial covers the zero-shot reference mapping with scGPT. This workflow achieves accurate and fast reference mapping for scRNA-seq datasets without fine-tuning (or any extensive training) of scGPT. To further boost the performance, we recommend fine-tuning scGPT.
We will use COVID-19 dataset to demonstrate the zero-shot reference mapping. You can download the processed reference and query datasets from here. The COVID-19 dataset is derived from the work by Lotfollahi et al, which contains 18 distinct batches and diverse samples from lung tissues. The reference dataset consists of 15,997 cells and the query dataset contains 4,003 cells. You may place the dataset under data directory in the outer level.
Particularly, we use the scGPT_human model to provide embeddings out of the box. You may download it from here.
The zero-shot reference mapping workflow is as follows:
1. Load and pre-process the dataset
2. Generate scGPT embeddings for each cell in reference and query datasets
3. Transfer the annotations from reference to query dataset
At the appendix of this tutorial, we will also showcase the zero-shot reference mapping on Lung dataset. You can find the dataset here.
We use a similarity-based method for transferring the annotation, which involves comparing the similarity between the query cell embedding and the reference cell embeddings. We use FAISS to perform the similarity search.
Open in Colab
# Specifically for Google Colab, install dependencies and download data
import os
import sys
if "google.colab" in sys.modules:
print("Running on Google Colab")
print("Installing dependencies...")
!pip install -U scgpt
0s completed at 12:03 PM
Haotian Cui


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Zero-shot reference mapping tutorial with scGPT
Introduction
This tutorial covers the zero-shot reference mapping with scGPT. This workflow achiev
datasets without fine-tuning (or any extensive training) of scGPT. To further boost the
We will use COVID-19 dataset to demonstrate the zero-shot reference mapping. You ca
from here. The COVID-19 dataset is derived from the work by Lotfollahi et al, which cor
tissues. The reference dataset consists of 15,997 cells and the query dataset contains
directory in the outer level.
Particularly, we use the scGPT_human model to provide embeddings out of the box. Yo
The zero-shot reference mapping workflow is as follows:
1. Load and pre-process the dataset
2. Generate scGPT embeddings for each cell in reference and query datasets
3. Transfer the annotations from reference to query dataset
At the appendix of this tutorial, we will also showcase the zero-shot reference mapping
We use a similarity-based method for transferring the annotation, which involves comp
and the reference cell embeddings. We use FAISS to perform the similarity search.
Open in Colab
Change runtime type
Runtime type
Python 3
Hardware accelerator
CPU T4 GPU A100 GPU V100 GPU
TPU
Want access to premium GPUs? Purchase additional compute units
Cancel Save
# Specifically for Google Colab, install dependencies and download data
import os
import sys
if "google.colab" in sys.modules:
print("Running on Google Colab")
print("Installing dependencies...")
!pip install -U scgpt

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

Zero-shot reference mapping tutorial with scGPT

Introduction

This tutorial covers the zero-shot reference mapping with scGPT. This workflow achieves accurate and fast reference mapping for scRNA-seq datasets without fine-tuning (or any extensive training) of scGPT. To further boost the performance, we recommend fine-tuning scGPT.

We will use COVID-19 dataset to demonstrate the zero-shot reference mapping. You can download the processed reference and query datasets from here. The COVID-19 dataset is derived from the work by Lotfollahi et al. which contains 18 distinct batches and diverse samples from lung tissues. The reference dataset consists of 15,997 cells and the query dataset contains 4,003 cells. You may place the dataset under `data` directory in the outer level.

Particularly, we use the `scGPT_human` model to provide embeddings out of the box. You may download it from here.

The zero-shot reference mapping workflow is as follows:

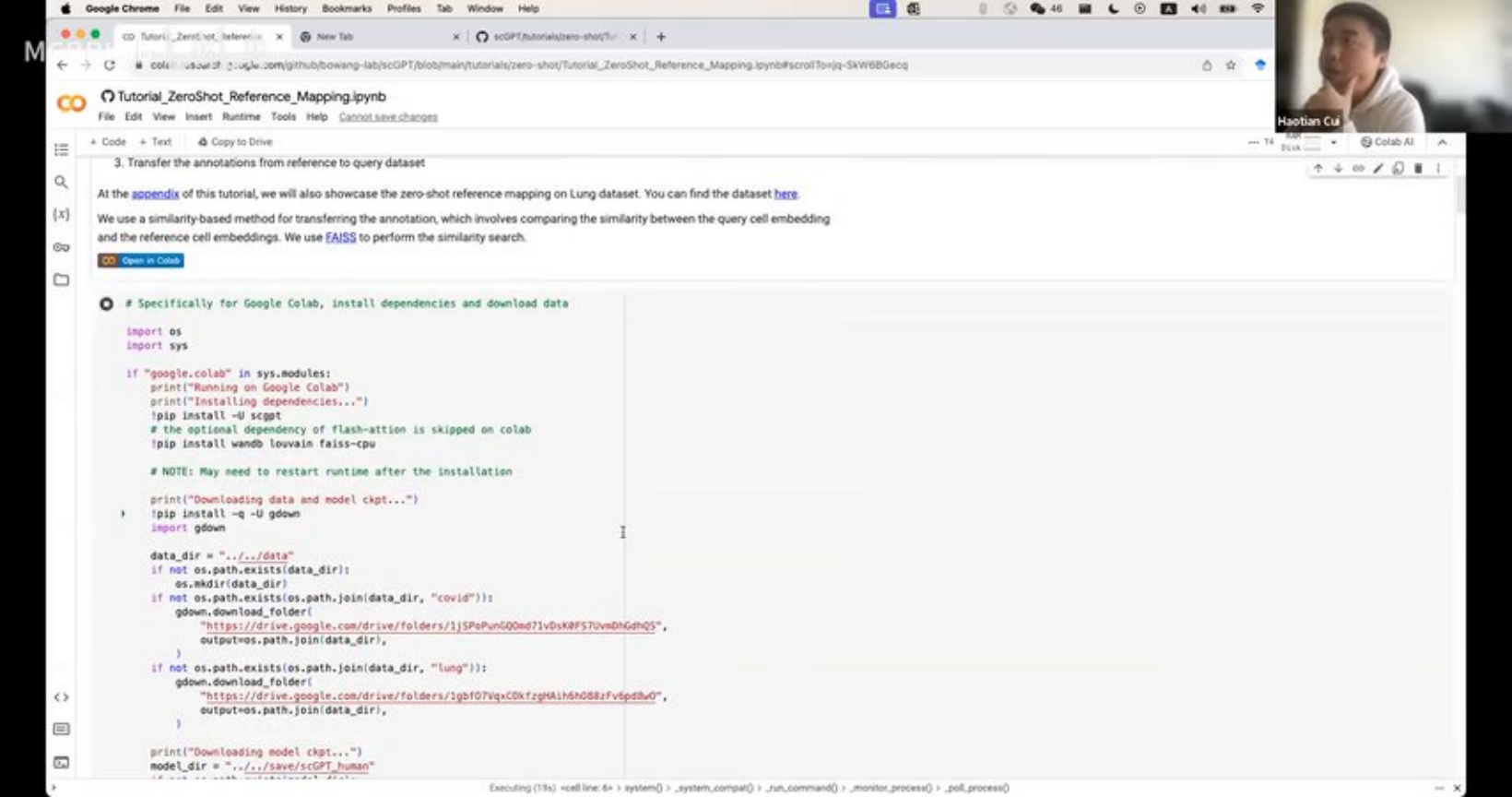
1. Load and pre-process the dataset
2. Generate scGPT embeddings for each cell in reference and query datasets
3. Transfer the annotations from reference to query dataset

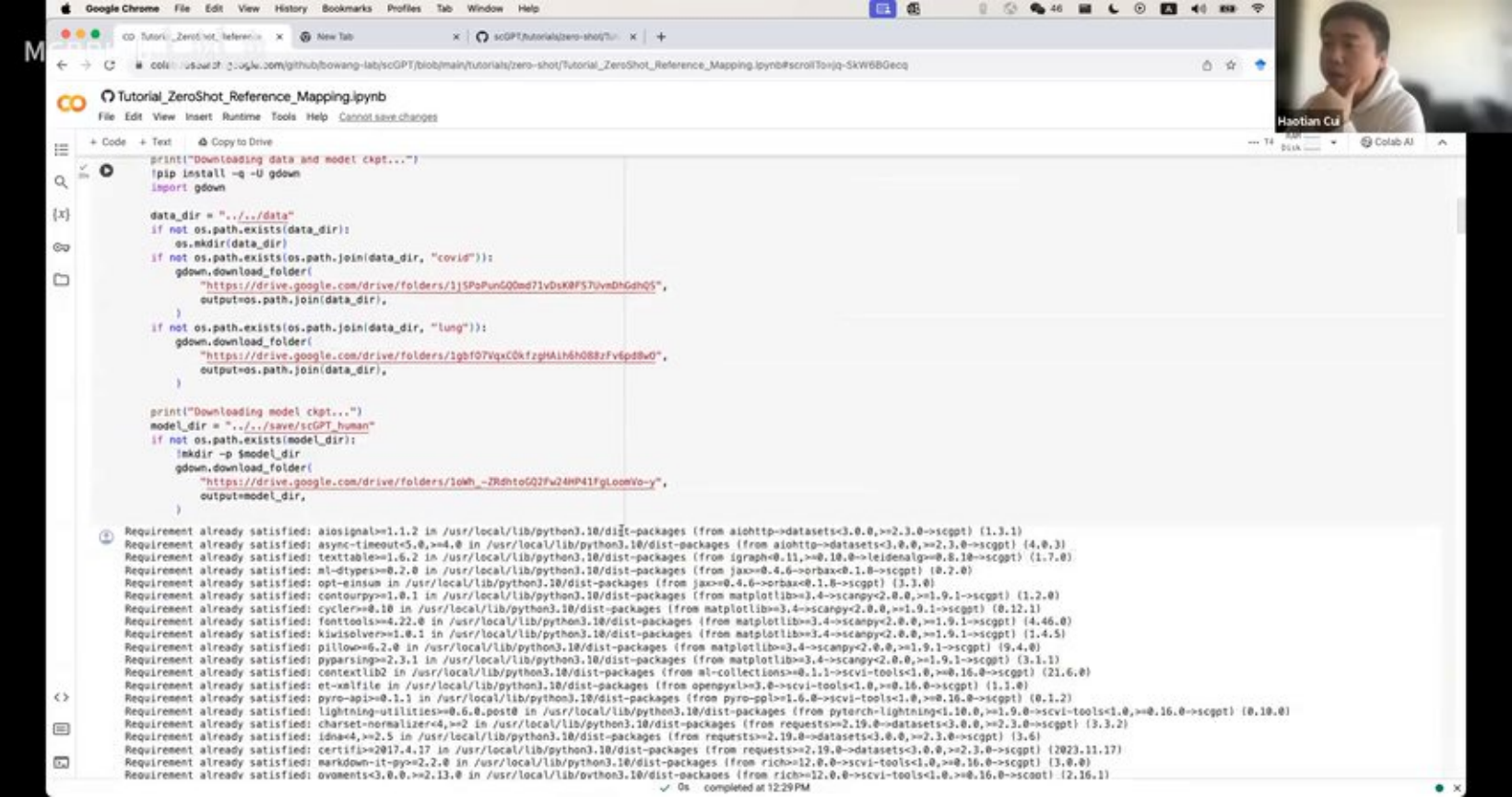
At the appendix of this tutorial, we will also showcase the zero-shot reference mapping on Lung dataset. You can find the dataset here.

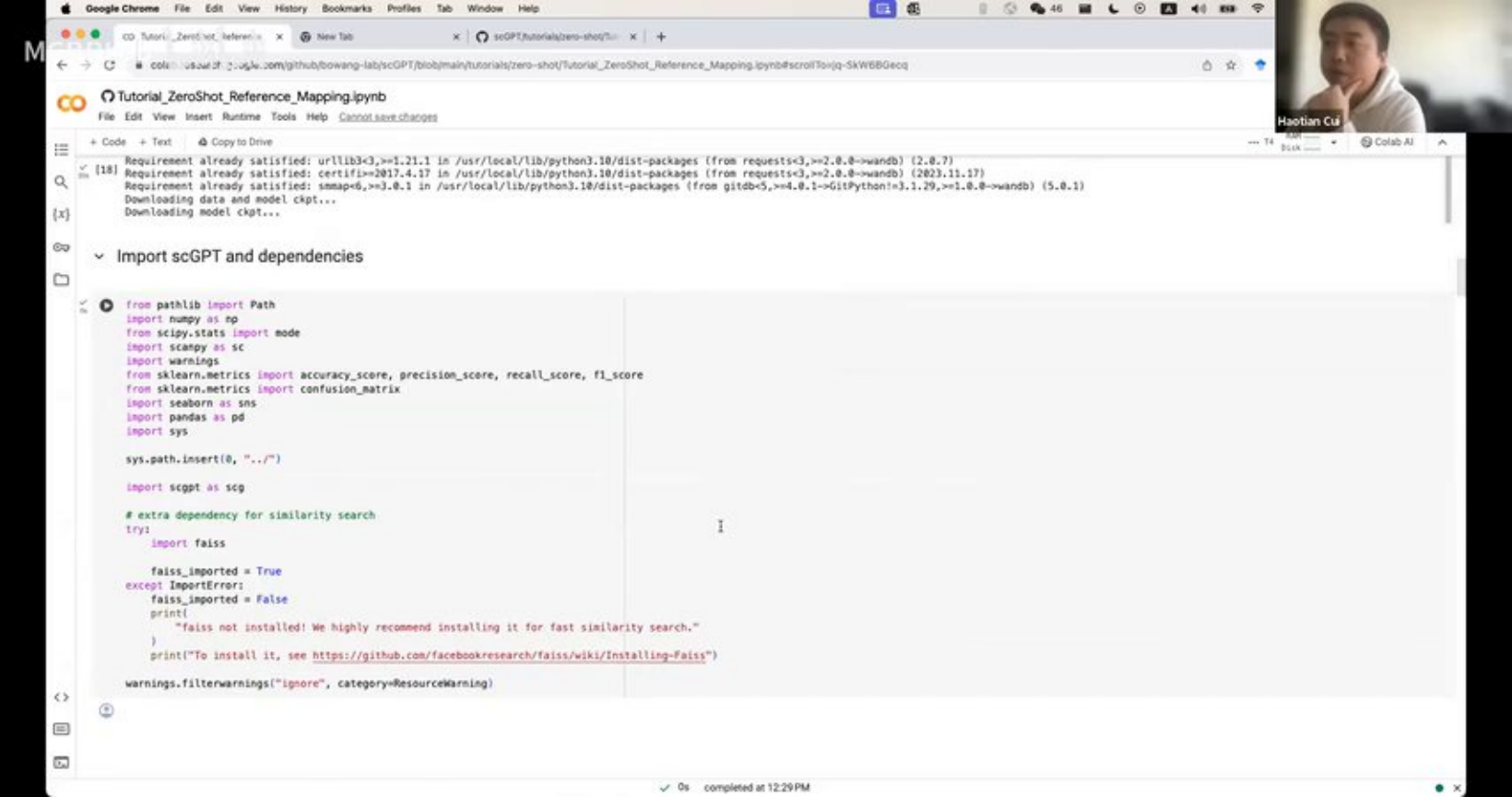
We use a similarity-based method for transferring the annotation, which involves comparing the similarity between the query cell embedding and the reference cell embeddings. We use FAISS to perform the similarity search.

Open in Colab

```
# Specifically for Google Colab, install dependencies and download data

import os
import sys

if "google.colab" in sys.modules:
    print("Running on Google Colab")
    print("Installing dependencies...")
    !pip install -U scgpt
    # the optional dependency of flash-attion is skipped on colab
    !pip install wandb louvain faiss-cpu
```


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
3. Transfer the annotations from reference to query dataset
At the appendix of this tutorial, we will also showcase the zero-shot reference mapping on Lung dataset. You can find the dataset here.
We use a similarity-based method for transferring the annotation, which involves comparing the similarity between the query cell embedding and the reference cell embeddings. We use FAISS to perform the similarity search.
Open in Colab
# Specifically for Google Colab, install dependencies and download data
import os
import sys
if "google.colab" in sys.modules:
print("Running on Google Colab")
print("Installing dependencies...")
!pip install -U scgpt
the optional dependency of flash-attion is skipped on colab
!pip install wandb louvain faiss-cpu
# NOTE: May need to restart runtime after the installation
print("Downloading data and model ckpt...")
!pip install -q -U gdown
import gdown
data_dir = "../../data"
if not os.path.exists(data_dir):
os.mkdir(data_dir)
if not os.path.exists(os.path.join(data_dir, "covid")):
gdown.download_folder(
"https://drive.google.com/drive/folders/1jSPoPunGQOmd71vDsK0FS7UvmDhGdhQS",
output=os.path.join(data_dir),
)
if not os.path.exists(os.path.join(data_dir, "lung")):
gdown.download_folder(
"https://drive.google.com/drive/folders/1gbfG7VqxC0kfzgHAih6hO88zFv6pd8wO",
output=os.path.join(data_dir),
)
print("Downloading model ckpt...")
model_dir = "../../save/scGPT_human"
Haotian Cui


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
print("Downloading data and model ckpt...")
!pip install -q -U gdown
import gdown
data_dir = "../../data"
if not os.path.exists(data_dir):
os.mkdir(data_dir)
if not os.path.exists(os.path.join(data_dir, "covid")):
gdown.download_folder(
"https://drive.google.com/drive/folders/1j5PoPunGQOmd71vDsK0FS7UvmDhGdhQ5",
output=os.path.join(data_dir),
)
if not os.path.exists(os.path.join(data_dir, "lung")):
gdown.download_folder(
"https://drive.google.com/drive/folders/1gbfO7VqxC0kfzgHAih6hO88zFv6pd8wO",
output=os.path.join(data_dir),
)
print("Downloading model ckpt...")
model_dir = "../../save/scGPT_human"
if not os.path.exists(model_dir):
!mkdir -p $model_dir
gdown.download_folder(
"https://drive.google.com/drive/folders/1oWh_-ZRdhtoGQ2Fw24HP41FgLoomVo-y",
output=model_dir,
)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets<3.0.0,>=2.3.0->scgpt) (1.3.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets<3.0.0,>=2.3.0->scgpt) (4.0.3)
Requirement already satisfied: texttable>=1.6.2 in /usr/local/lib/python3.10/dist-packages (from igraph<0.11,>=0.10.0->leidenalg>=0.8.10->scgpt) (1.7.0)
Requirement already satisfied: ml-dtypes>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from jax>=0.4.6->orbax<0.1.8->scgpt) (0.2.0)
Requirement already satisfied: opt-einsum in /usr/local/lib/python3.10/dist-packages (from jax>=0.4.6->orbax<0.1.8->scgpt) (3.3.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.4->scanpy<2.0.0,>=1.9.1->scgpt) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.4->scanpy<2.0.0,>=1.9.1->scgpt) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.4->scanpy<2.0.0,>=1.9.1->scgpt) (4.46.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.4->scanpy<2.0.0,>=1.9.1->scgpt) (1.4.5)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.4->scanpy<2.0.0,>=1.9.1->scgpt) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.4->scanpy<2.0.0,>=1.9.1->scgpt) (3.1.1)
Requirement already satisfied: contextlib2 in /usr/local/lib/python3.10/dist-packages (from ml-collections>=0.1.1->scvi-tools<1.0,>=0.16.0->scgpt) (21.6.0)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.10/dist-packages (from openpyxl>=3.0->scvi-tools<1.0,>=0.16.0->scgpt) (1.1.0)
Requirement already satisfied: pyro-api>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from pyro-ppl>=1.6.0->scvi-tools<1.0,>=0.16.0->scgpt) (0.1.2)
Requirement already satisfied: lightning-utilities>=0.6.0.post0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning<1.10.0,>=1.9.0->scvi-tools<1.0,>=0.16.0->scgpt) (0.10.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets<3.0.0,>=2.3.0->scgpt) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets<3.0.0,>=2.3.0->scgpt) (3.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets<3.0.0,>=2.3.0->scgpt) (2023.11.17)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=12.0.0->scvi-tools<1.0,>=0.16.0->scgpt) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=12.0.0->scvi-tools<1.0,>=0.16.0->scgpt) (2.16.1)
0s completed at 12:29 PM
Haotian Cui


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
Haotian Cui
[18] Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.0.0->wandb) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.0.0->wandb) (2023.11.17)
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from gitdb<5,>=4.0.1->GitPython!=3.1.29,>=1.0.0->wandb) (5.0.1)
Downloading data and model ckpt...
Downloading model ckpt...
Import scGPT and dependencies
from pathlib import Path
import numpy as np
from scipy.stats import mode
import scanpy as sc
import warnings
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix
import seaborn as sns
import pandas as pd
import sys
sys.path.insert(0, "../")
import scgpt as scg
# extra dependency for similarity search
try:
import faiss
faiss_imported = True
except ImportError:
faiss_imported = False
print(
"faiss not installed! We highly recommend installing it for fast similarity search."
)
print("To install it, see https://github.com/facebookresearch/faiss/wiki/Installing-Faiss")
warnings.filterwarnings("ignore", category=ResourceWarning)
0s completed at 12:29 PM

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

[19]

Load the dataset, you may download the dataset from here. We set the columns storing gene name columns, batch key and cell type key.

```
model_dir = Path("../../save/scGPT_human")
adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_train.h5ad')
test_adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_test.h5ad')

cell_type_key = "celltype"
gene_col = "gene_name"
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
```

Embed the reference dataset

```
ref_embed_adata = scg.tasks.embed_data(
    adata,
    model_dir,
    gene_col=gene_col,
    batch_size=64,
)
```

```
scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
```

0s completed at 12:29 PM

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

[19]

Load the dataset, you may download the dataset from here. We set the columns storing gene name columns, batch key and cell type key.

```
model_dir = Path("../../ ../data/covid/batch_covid_subsampled_train.h5ad (cmd + click)
adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_train.h5ad')
test_adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_test.h5ad')

cell_type_key = "celltype"
gene_col = "gene_name"
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
```

Embed the reference dataset

```
ref_embed_adata = scg.tasks.embed_data(
    adata,
    model_dir,
    gene_col=gene_col,
    batch_size=64,
)
```

```
scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
  warnings.warn(
Embedding cells:  94%|[redacted]| 236/250 [00:32<00:01,  7.38it/s]
```

0s completed at 12:29 PM

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

[19]

Load the dataset, you may download the dataset from here. We set the columns storing gene name columns, batch key and cell type key.

```
model_dir = Path("../../save/scGPT_human")
adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_train.h5ad')
test_adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_test.h5ad')

cell_type_key = "celltype"
gene_col = "gene_name"
```

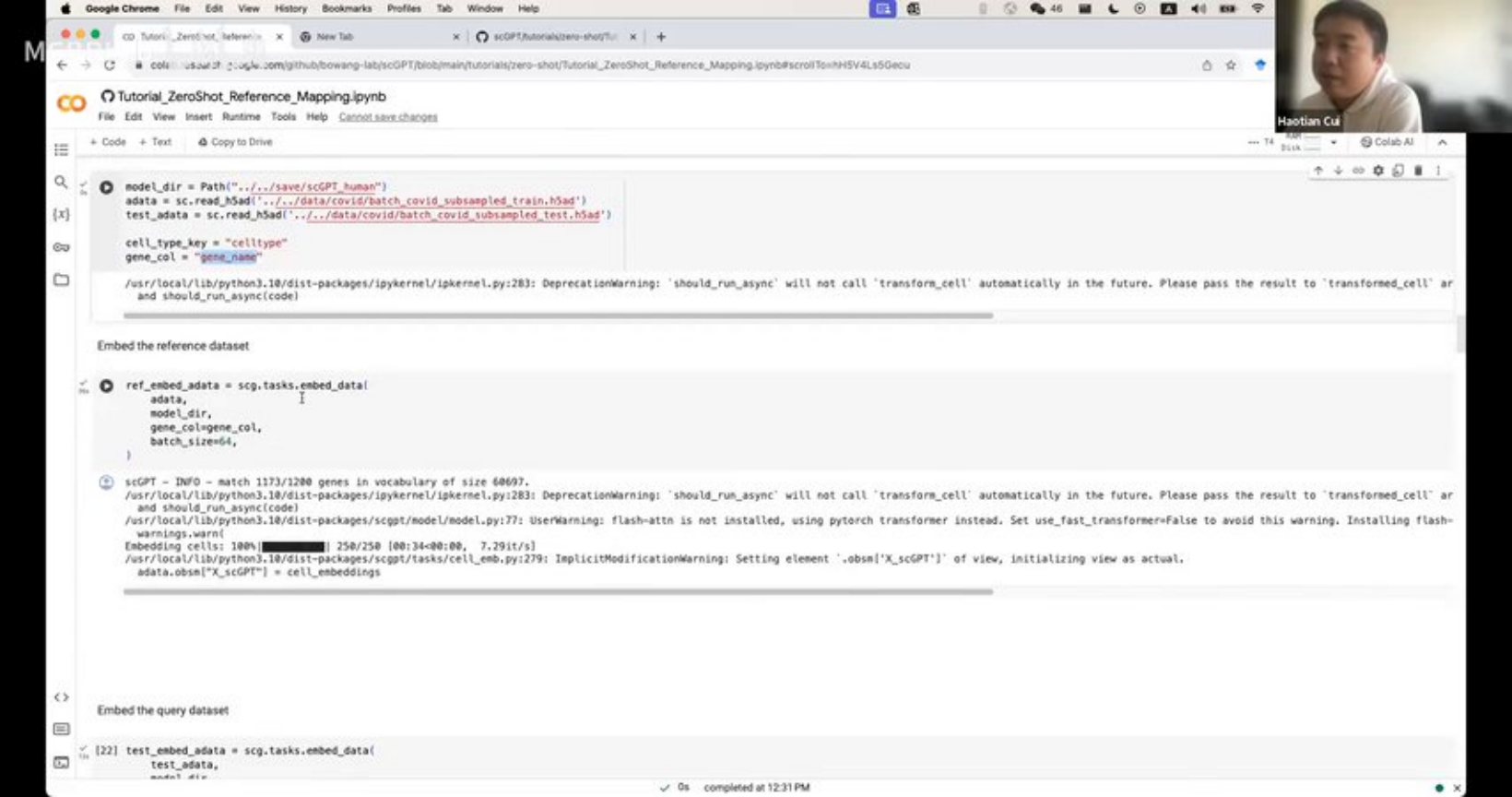
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
```

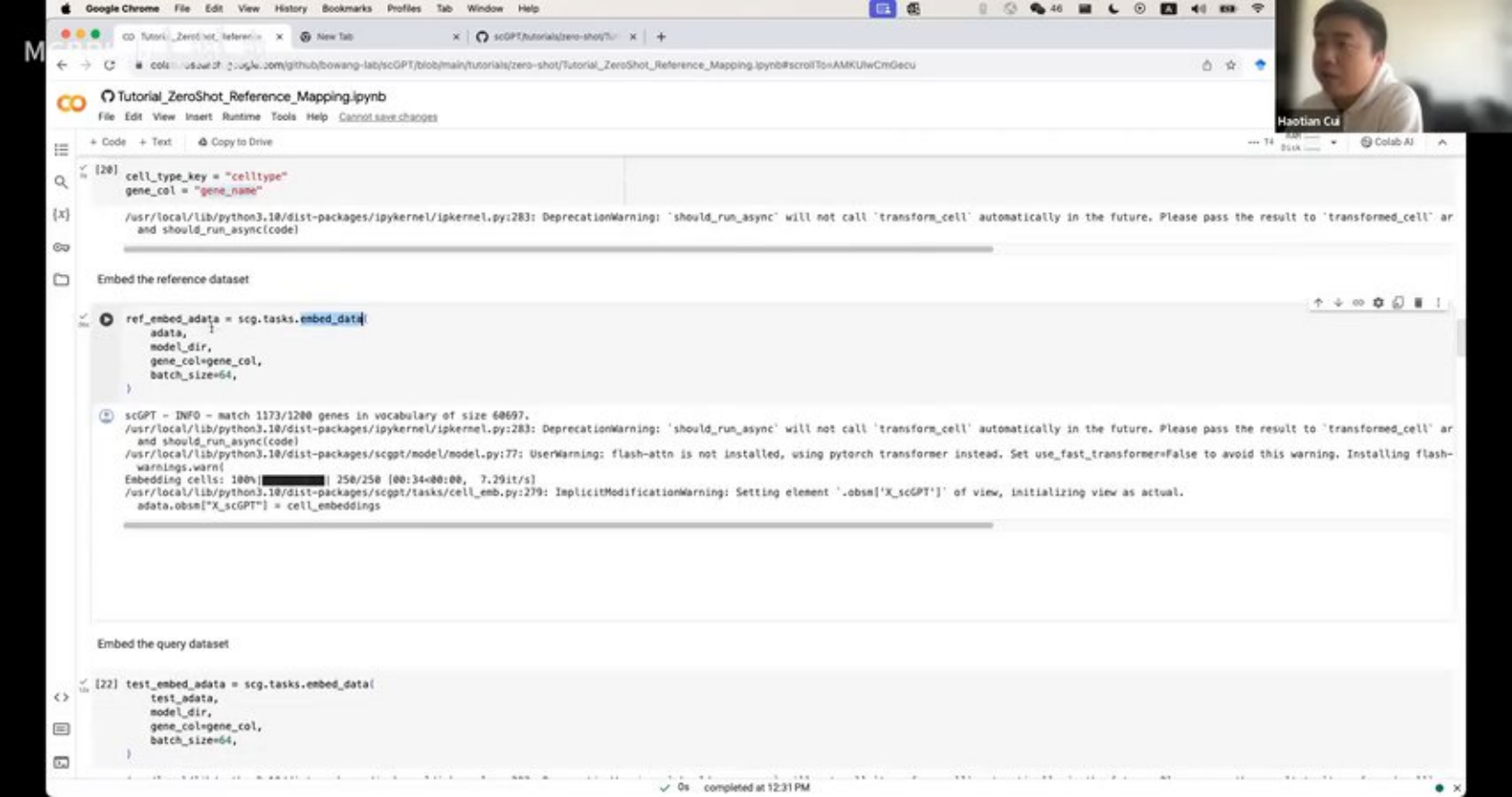
Embed the reference dataset

[21]
```
ref_embed_adata = scg.tasks.embed_data(
    adata,
    model_dir,
    gene_col=gene_col,
    batch_size=64,
)
```

```
scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
  warnings.warn(
Embedding cells: 100%|██████████| 250/250 [00:34<00:00,  7.29it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element `.obsm['X_scGPT']` of view, initializing view as actual.
  adata.obsm["X_scGPT"] = cell_embeddings
```

0s completed at 12:30 PM


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
model_dir = Path("../../save/scGPT_human")
adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_train.h5ad')
test_adata = sc.read_h5ad('../../data/covid/batch_covid_subsampled_test.h5ad')
cell_type_key = "celltype"
gene_col = "gene_name"
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
Embed the reference dataset
ref_embed_adata = scg.tasks.embed_data(
adata,
model_dir,
gene_col=gene_col,
batch_size=64,
)
scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
warnings.warn(
Embedding cells: 100%| 250/250 [00:34<00:00, 7.29it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element `.obsm['X_scGPT']` of view, initializing view as actual.
adata.obsm["X_scGPT"] = cell_embeddings
Embed the query dataset
[22] test_embed_adata = scg.tasks.embed_data(
test_adata,
0s completed at 12:31 PM
Haotian Cui


Haotian Cui
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
[20] cell_type_key = "celltype"
gene_col = "gene_name"
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
Embed the reference dataset
ref_embed_adata = scg.tasks.embed_data(
adata,
model_dir,
gene_col=gene_col,
batch_size=64,
)
scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
warnings.warn(
Embedding cells: 100%| | 250/250 [00:34<00:00, 7.29it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element `.obsm['X_scGPT']` of view, initializing view as actual.
adata.obsm["X_scGPT"] = cell_embeddings
Embed the query dataset
[22] test_embed_adata = scg.tasks.embed_data(
test_adata,
model_dir,
gene_col=gene_col,
batch_size=64,
)
0s completed at 12:31 PM

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
  warnings.warn(
Embedding cells: 100%|██████████| 250/250 [00:34<00:00,  7.29it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element `.obsm['X_scGPT']` of view, initializing view as actual.
  adata.obsm["X_scGPT"] = cell_embeddings
```

Embed the query dataset

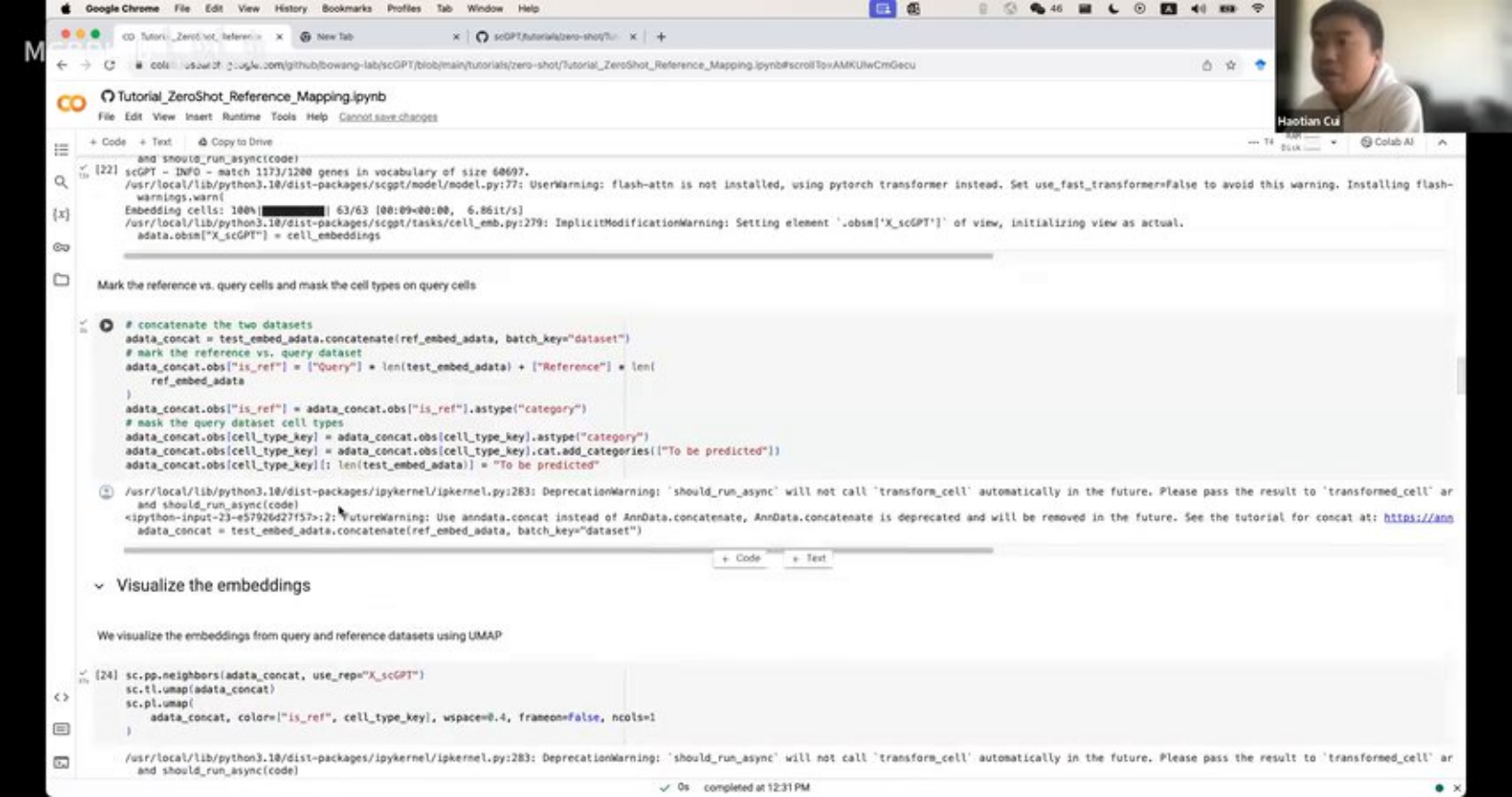
```
test_embed_adata = scg.tasks.embed_data(
    test_adata,
    model_dir,
    gene_col=gene_col,
    batch_size=64,
)
```

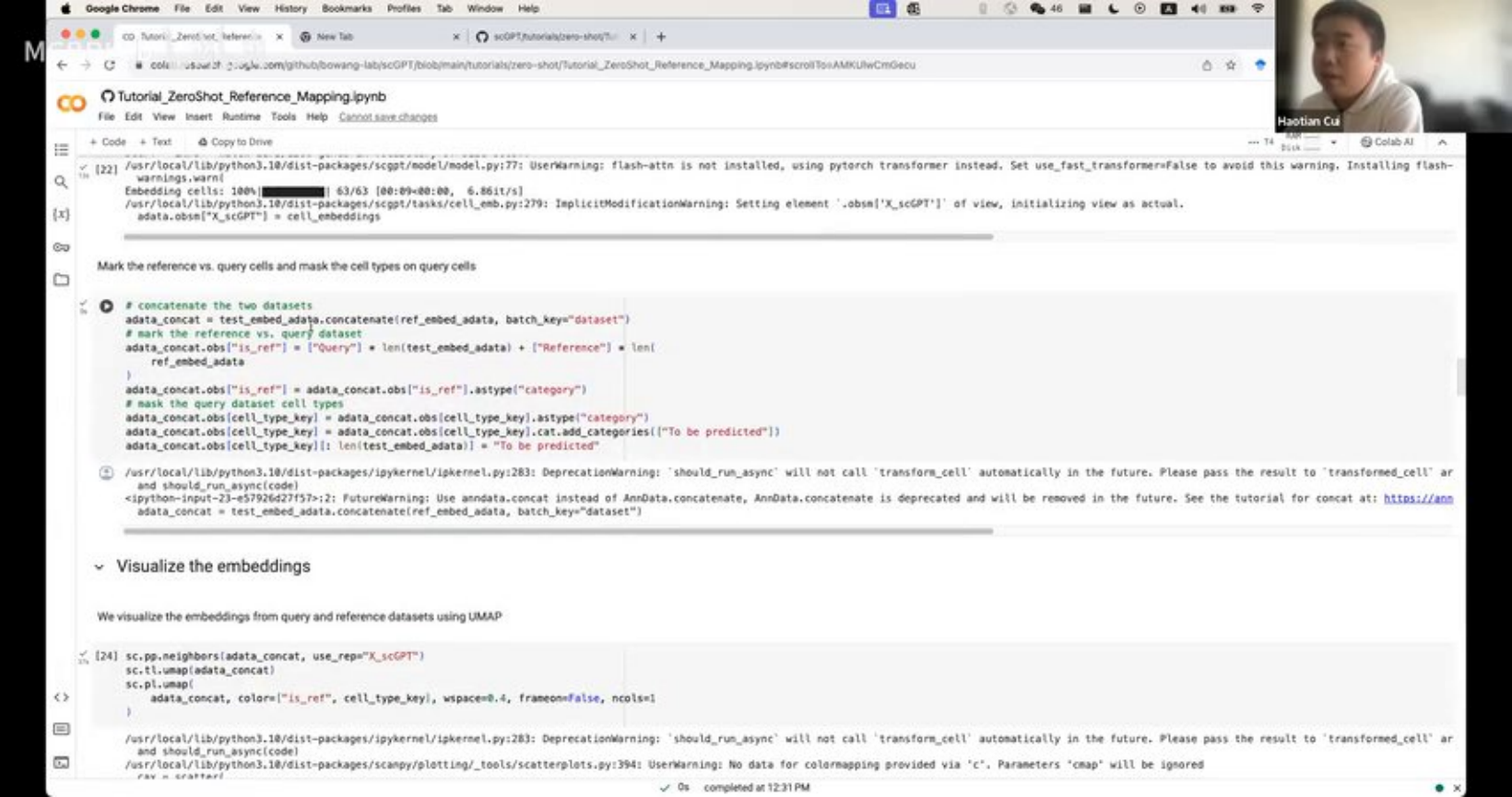
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
  warnings.warn(
Embedding cells: 100%|██████████| 63/63 [00:09<00:00,  6.86it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element `.obsm['X_scGPT']` of view, initializing view as actual.
  adata.obsm["X_scGPT"] = cell_embeddings
```

Mark the reference vs. query cells and mask the cell types on query cells

```
[23] # concatenate the two datasets
     adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
     # mark the reference vs. query dataset
```

0s completed at 12:31 PM


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
and should_run_async(code)
[22] scGPT - INFO - match 1173/1200 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
warnings.warn(
Embedding cells: 100%| 63/63 [00:09<00:00, 6.86it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element `.obsm['X_scGPT']` of view, initializing view as actual.
adata.obsm["X_scGPT"] = cell_embeddings
Mark the reference vs. query cells and mask the cell types on query cells
# concatenate the two datasets
adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
# mark the reference vs. query dataset
adata_concat.obs["is_ref"] = ["Query"] * len(test_embed_adata) + ["Reference"] * len(
ref_embed_adata
)
adata_concat.obs["is_ref"] = adata_concat.obs["is_ref"].astype("category")
# mask the query dataset cell types
adata_concat.obs[cell_type_key] = adata_concat.obs[cell_type_key].astype("category")
adata_concat.obs[cell_type_key] = adata_concat.obs[cell_type_key].cat.add_categories(["To be predicted"])
adata_concat.obs[cell_type_key][: len(test_embed_adata)] = "To be predicted"
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
<ipython-input-23-e57926d27f57>:2: FutureWarning: Use anndata.concat instead of AnnData.concatenate, AnnData.concatenate is deprecated and will be removed in the future. See the tutorial for concat at: https://ann
adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
+ Code + Text
Visualize the embeddings
We visualize the embeddings from query and reference datasets using UMAP
[24] sc.pp.neighbors(adata_concat, use_rep="X_scGPT")
sc.tl.umap(adata_concat)
sc.pl.umap(
adata_concat, color=["is_ref", cell_type_key], wspace=0.4, frameon=False, ncols=1
)
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
0s completed at 12:31 PM
Haotian Cui


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
[22] /usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
warnings.warn(
Embedding cells: 100%| 63/63 [00:09<00:00, 6.86it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element '.obsm['X_scGPT']' of view, initializing view as actual.
adata.obsm["X_scGPT"] = cell_embeddings
Mark the reference vs. query cells and mask the cell types on query cells
# concatenate the two datasets
adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
# mark the reference vs. query dataset
adata_concat.obs["is_ref"] = ["Query"] * len(test_embed_adata) + ["Reference"] * len(
ref_embed_adata
)
adata_concat.obs["is_ref"] = adata_concat.obs["is_ref"].astype("category")
# mask the query dataset cell types
adata_concat.obs[cell_type_key] = adata_concat.obs[cell_type_key].astype("category")
adata_concat.obs[cell_type_key] = adata_concat.obs[cell_type_key].cat.add_categories(["To be predicted"])
adata_concat.obs[cell_type_key][: len(test_embed_adata)] = "To be predicted"
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
<ipython-input-23-e57926d27f57>:2: FutureWarning: Use anndata.concat instead of AnnData.concatenate, AnnData.concatenate is deprecated and will be removed in the future. See the tutorial for concat at: https://ann
adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
Visualize the embeddings
We visualize the embeddings from query and reference datasets using UMAP
[24] sc.pp.neighbors(adata_concat, use_rep="X_scGPT")
sc.tl.umap(adata_concat)
sc.pl.umap(
adata_concat, color=["is_ref", cell_type_key], wspace=0.4, frameon=False, ncols=1
)
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
0s completed at 12:31 PM
Haotian Cui

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
adata_concat.obs[cell_type_key] = adata_concat.obs[cell_type_key].cat.add_categories(["To be predicted"])
adata_concat.obs[cell_type_key][: len(test_embed_adata)] = "To be predicted"
```

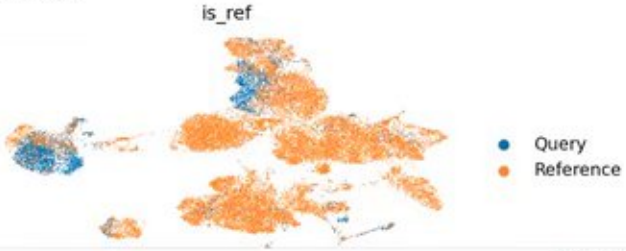
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
<ipython-input-23-e57926d27f57>:2: FutureWarning: Use anndata.concat instead of AnnData.concatenate, AnnData.concatenate is deprecated and will be removed in the future. See the tutorial for concat at: https://ann
  adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
```

Visualize the embeddings

We visualize the embeddings from query and reference datasets using UMAP

```
[24] sc.pp.neighbors(adata_concat, use_rep="X_scGPT")
     sc.tl.umap(adata_concat)
     sc.pl.umap(
         adata_concat, color=["is_ref", cell_type_key], wspace=0.4, frameon=False, ncols=1
     )
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  cax = scatter(
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  cax = scatter(
```



0s completed at 12:31 PM

Haotian Cui

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

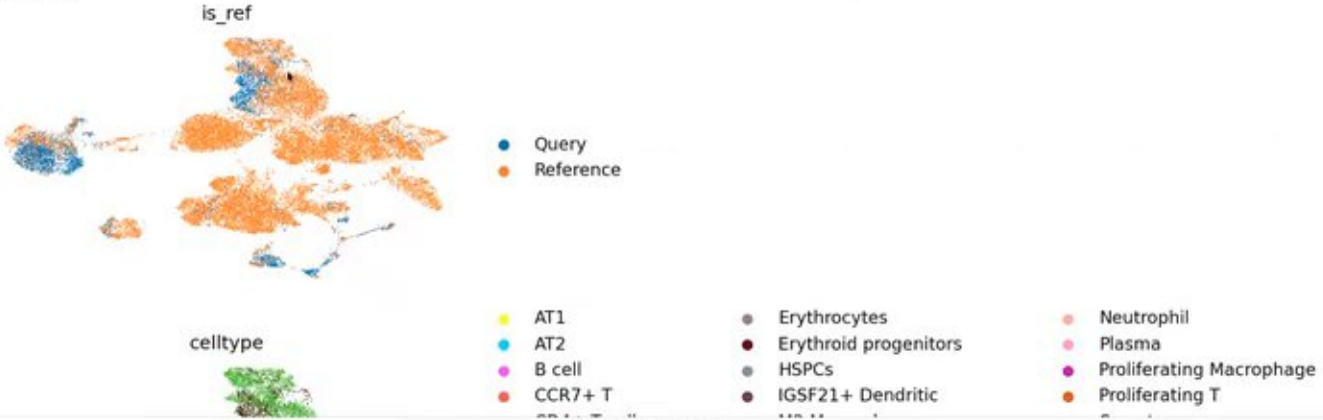
+ Code + Text Copy to Drive

Colab AI

We visualize the embeddings from query and reference datasets using UMAP

```
sc.pp.neighbors(adata_concat, use_rep="X_scGPT")
sc.tl.umap(adata_concat)
sc.pl.umap(
    adata_concat, color=["is_ref", cell_type_key], wspace=0.4, frameon=False, ncols=1
)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  cax = scatter(
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  cax = scatter(
```



0s completed at 12:31 PM

Haotian Cui

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

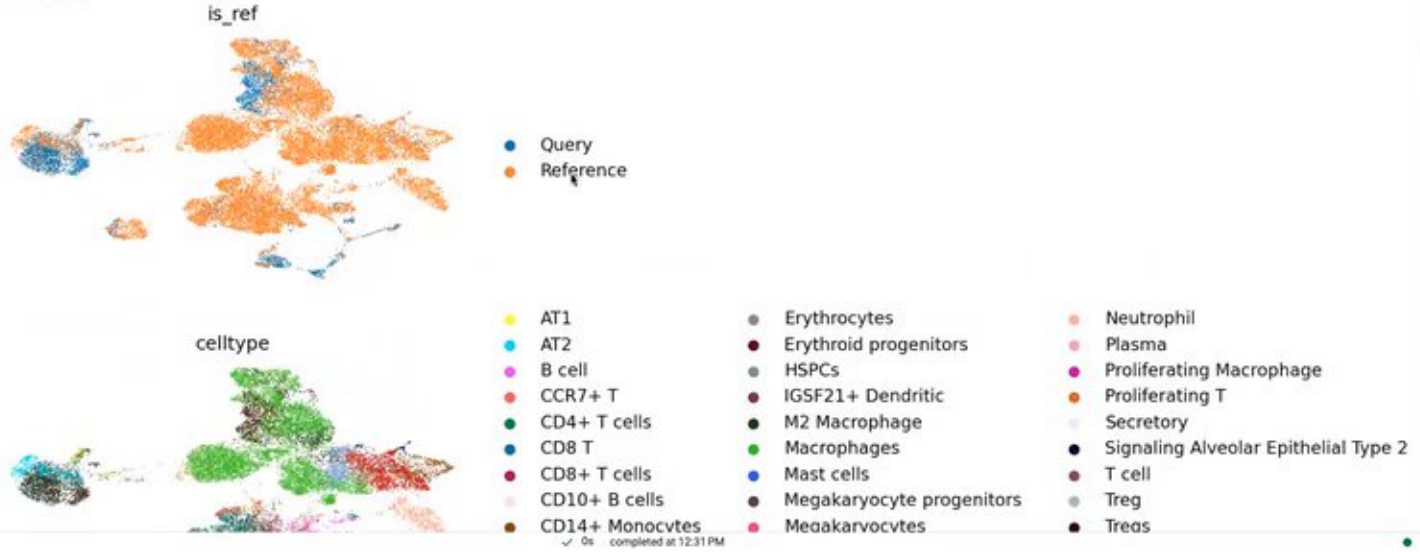
Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
sc.pl.umap(
    adata_concat, color=["is_ref", cell_type_key], wspace=0.4, frameon=False, ncols=1
)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  cax = scatter(
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  cax = scatter(
```



0s completed at 12:31 PM

Haotian Cui

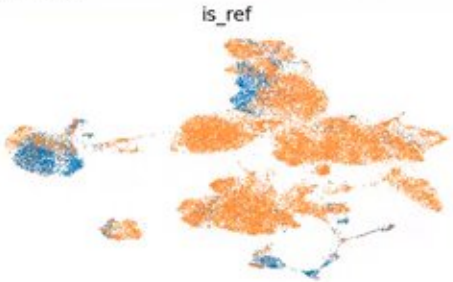
Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Tutorial_ZeroShot_Reference_Mapping.ipynb

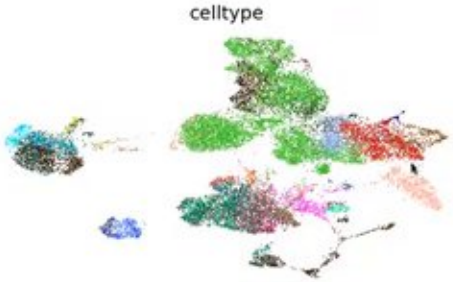
File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
    cax = scatter(
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
    cax = scatter(
```



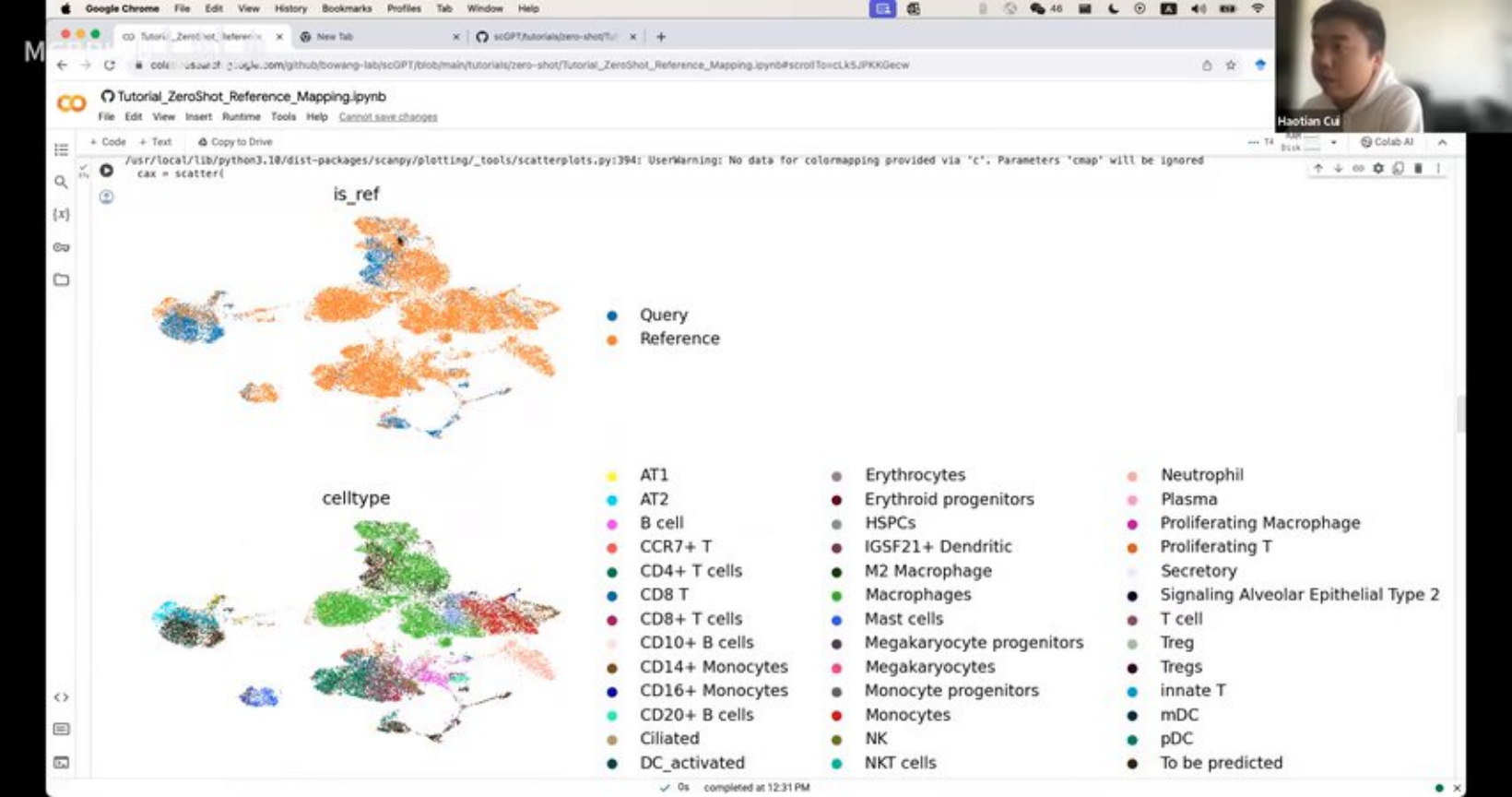
- Query
- Reference

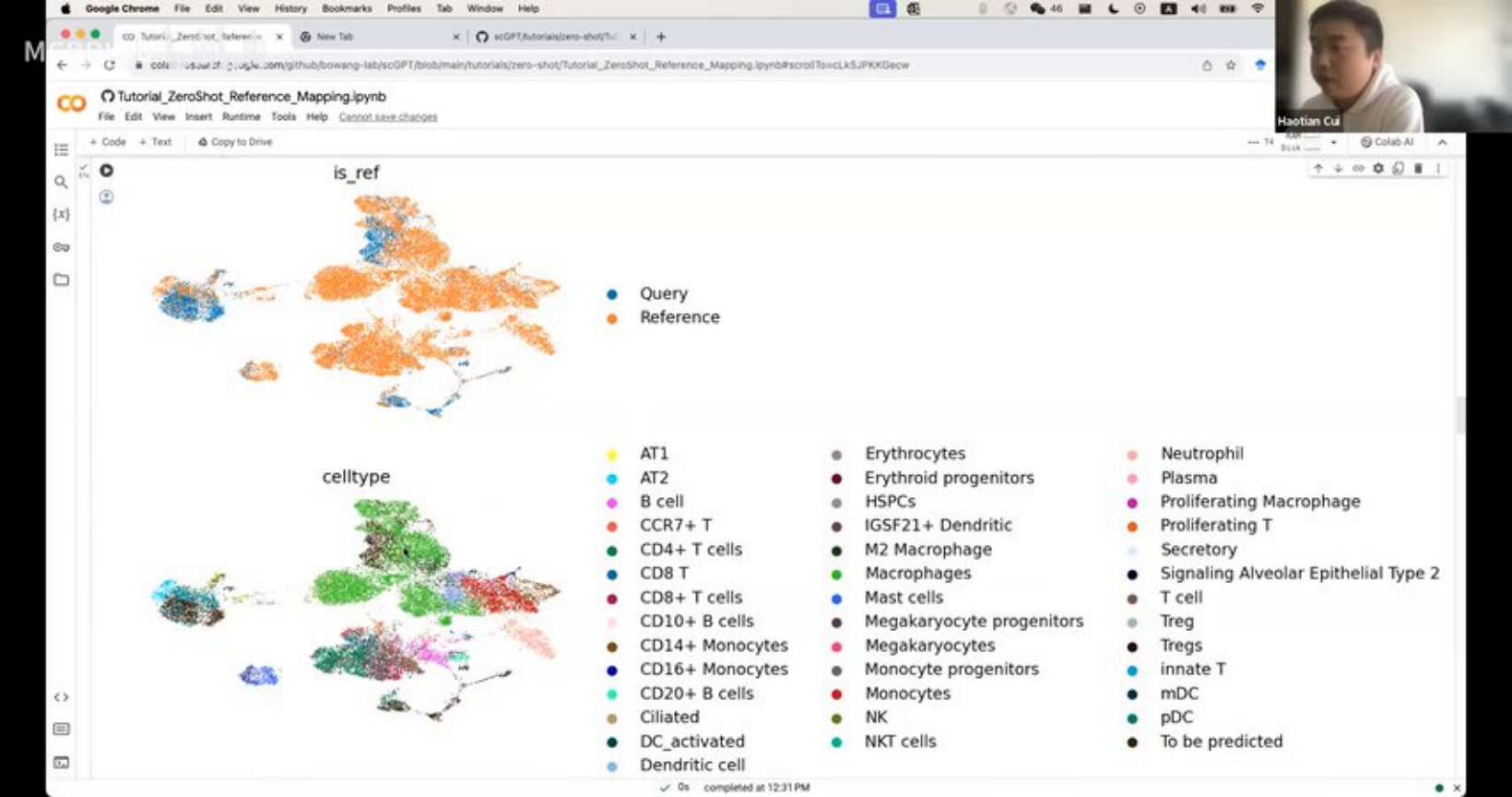


- AT1
- AT2
- B cell
- CCR7+ T
- CD4+ T cells
- CD8 T
- CD8+ T cells
- CD10+ B cells
- CD14+ Monocytes
- CD16+ Monocytes
- CD20+ B cells
- Ciliated
- DC_activated
- Erythrocytes
- Erythroid progenitors
- HSPCs
- IGSF21+ Dendritic
- M2 Macrophage
- Macrophages
- Mast cells
- Megakaryocyte progenitors
- Megakaryocytes
- Monocyte progenitors
- Monocytes
- NK
- NKT cells
- Neutrophil
- Plasma
- Proliferating Macrophage
- Proliferating T
- Secretory
- Signaling Alveolar Epithelial Type 2
- T cell
- Treg
- Tregs
- innate T
- mDC
- pDC
- To be predicted

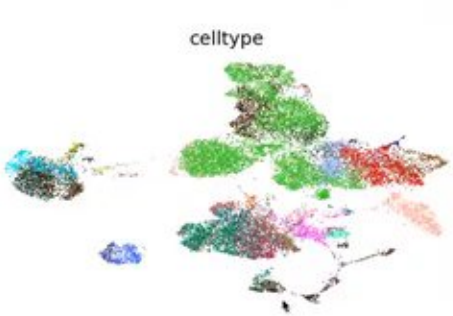
0s completed at 12:31 PM

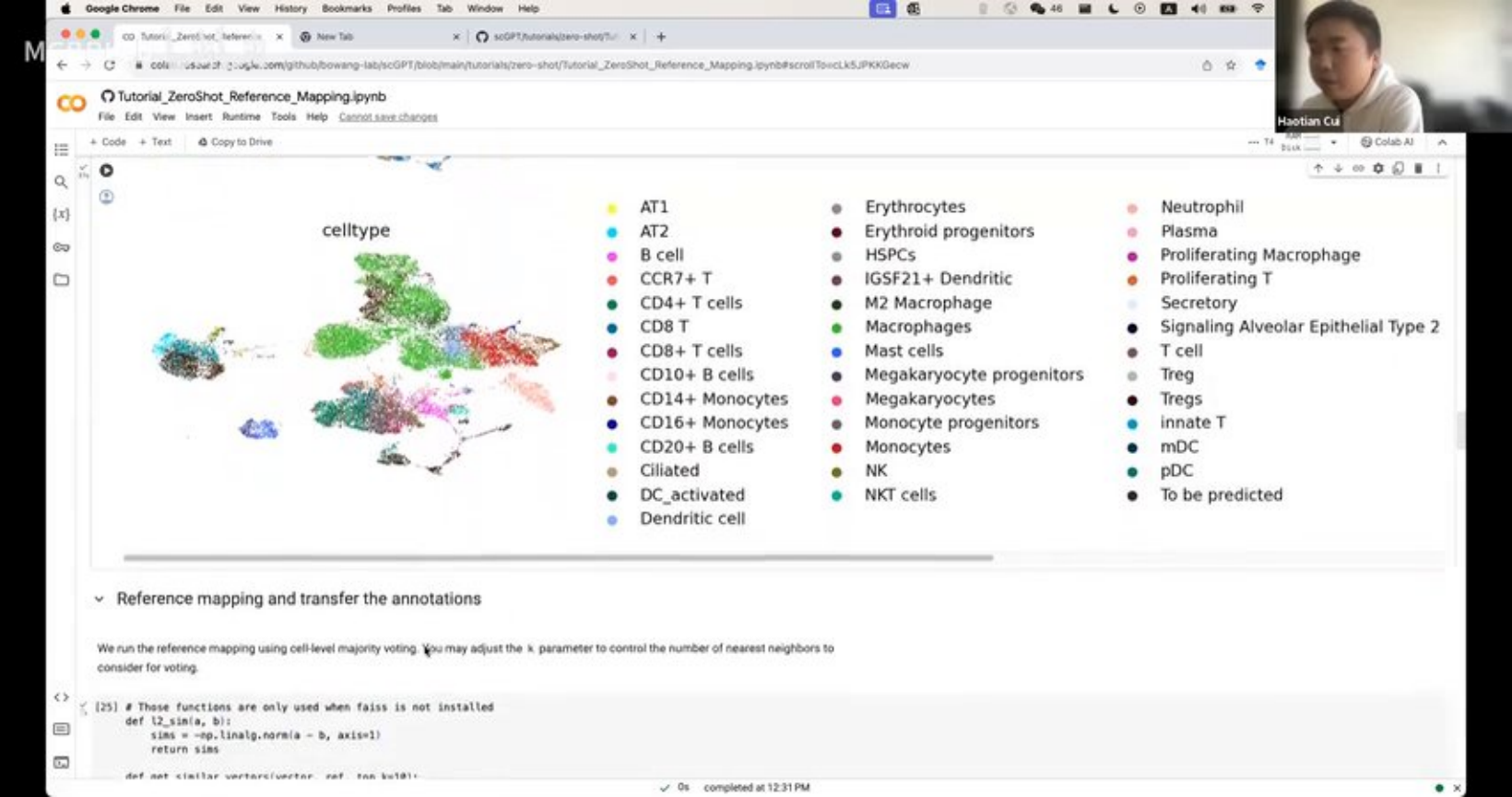
Haotian Cui

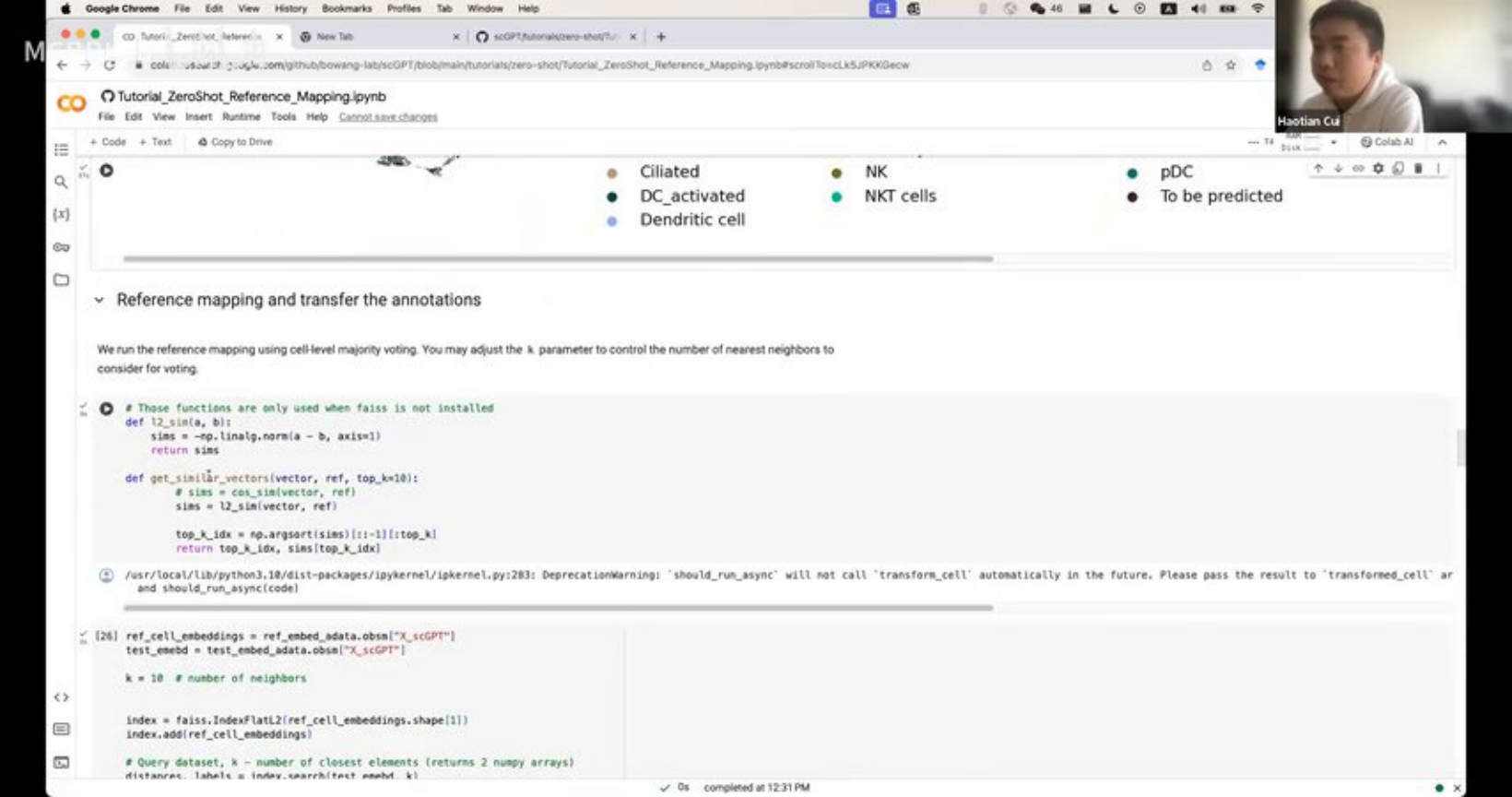

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
/usr/local/lib/python3.10/dist-packages/scanpy/plotting/_tools/scatterplots.py:394: UserWarning: No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
cax = scatter(
is_ref
Query
Reference
celltype
AT1
AT2
B cell
CCR7+ T
CD4+ T cells
CD8 T
CD8+ T cells
CD10+ B cells
CD14+ Monocytes
CD16+ Monocytes
CD20+ B cells
Ciliated
DC_activated
Erythrocytes
Erythroid progenitors
HSPCs
IGSF21+ Dendritic
M2 Macrophage
Macrophages
Mast cells
Megakaryocyte progenitors
Megakaryocytes
Monocyte progenitors
Monocytes
NK
NKT cells
Neutrophil
Plasma
Proliferating Macrophage
Proliferating T
Secretory
Signaling Alveolar Epithelial Type 2
T cell
Treg
Tregs
innate T
mDC
pDC
To be predicted
0s completed at 12:31 PM
Haotian Cui

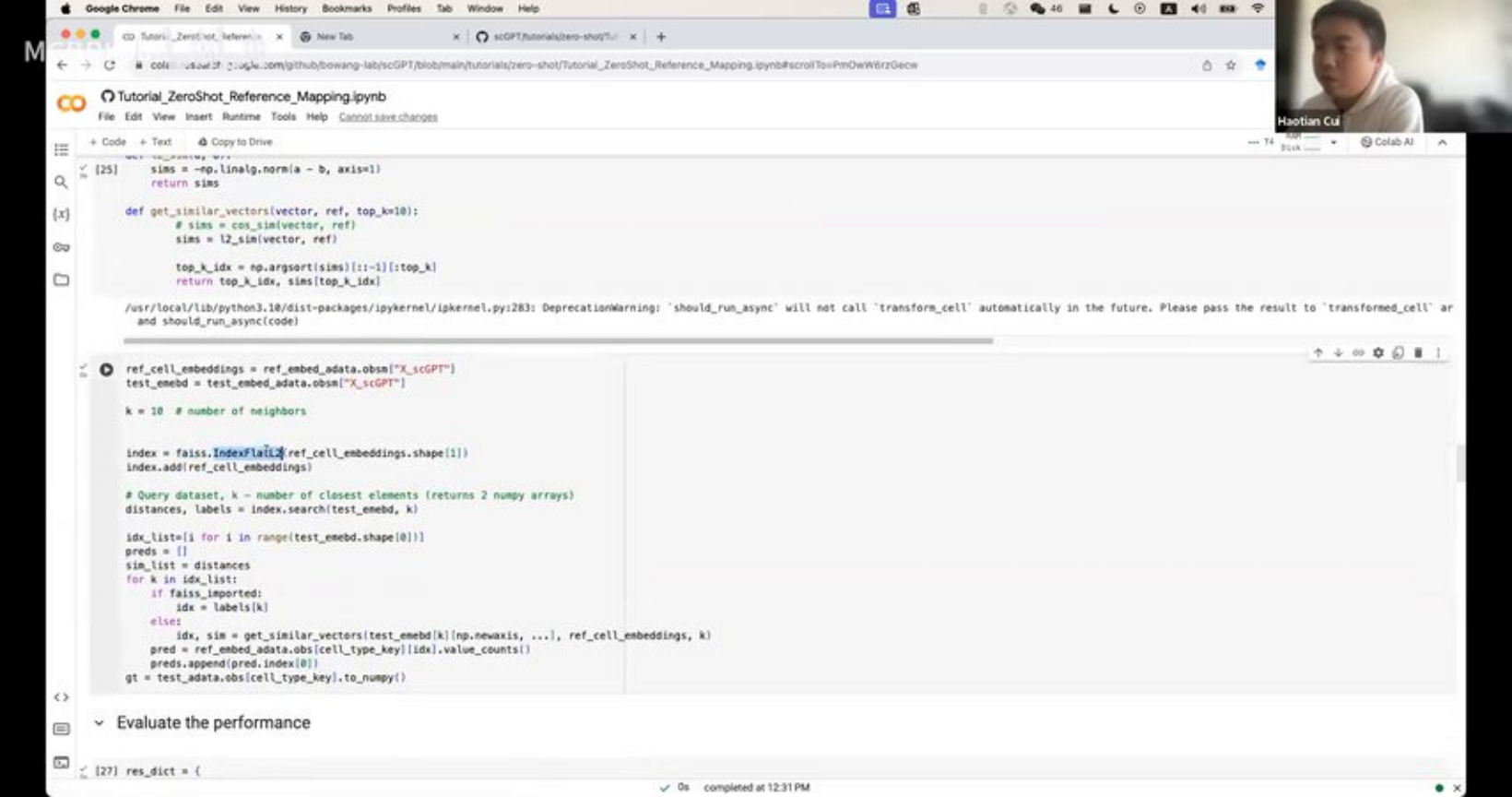

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Haotian Cui
is_ref
Query
Reference
celltype
AT1
AT2
B cell
CCR7+ T
CD4+ T cells
CD8 T
CD8+ T cells
CD10+ B cells
CD14+ Monocytes
CD16+ Monocytes
CD20+ B cells
Ciliated
DC_activated
Dendritic cell
Erythrocytes
Erythroid progenitors
HSPCs
IGSF21+ Dendritic
M2 Macrophage
Macrophages
Mast cells
Megakaryocyte progenitors
Megakaryocytes
Monocyte progenitors
Monocytes
NK
NKT cells
Neutrophil
Plasma
Proliferating Macrophage
Proliferating T
Secretory
Signaling Alveolar Epithelial Type 2
T cell
Treg
Tregs
innate T
mDC
pDC
To be predicted
0s completed at 12:31 PM

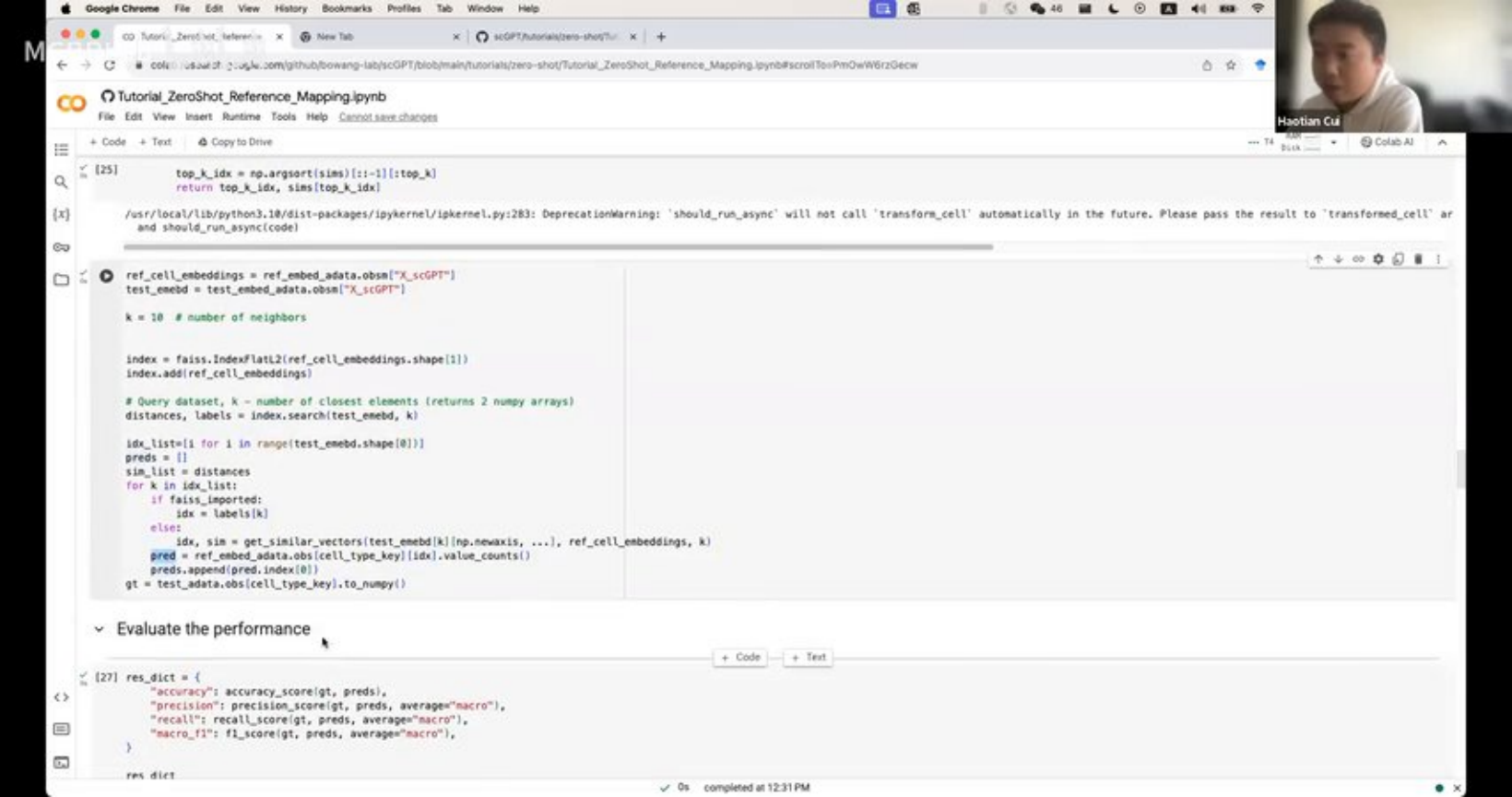

Query
Reference


celltype
AT1
AT2
B cell
CCR7+ T
CD4+ T cells
CD8 T
CD8+ T cells
CD10+ B cells
CD14+ Monocytes
CD16+ Monocytes
CD20+ B cells
Ciliated
DC_activated
Dendritic cell
Erythrocytes
Erythroid progenitors
HSPCs
IGSF21+ Dendritic
M2 Macrophage
Macrophages
Mast cells
Megakaryocyte progenitors
Megakaryocytes
Monocyte progenitors
Monocytes
NK
NKT cells
Neutrophil
Plasma
Proliferating Macrophage
Proliferating T
Secretory
Signaling Alveolar Epithelial Type 2
T cell
Treg
Tregs
innate T
mDC
pDC
To be predicted


Haotian Cui
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
celltype
AT1
AT2
B cell
CCR7+ T
CD4+ T cells
CD8 T
CD8+ T cells
CD10+ B cells
CD14+ Monocytes
CD16+ Monocytes
CD20+ B cells
Ciliated
DC_activated
Dendritic cell
Erythrocytes
Erythroid progenitors
HSPCs
IGSF21+ Dendritic
M2 Macrophage
Macrophages
Mast cells
Megakaryocyte progenitors
Megakaryocytes
Monocyte progenitors
Monocytes
NK
NKT cells
Neutrophil
Plasma
Proliferating Macrophage
Proliferating T
Secretory
Signaling Alveolar Epithelial Type 2
T cell
Treg
Tregs
innate T
mDC
pDC
To be predicted
Reference mapping and transfer the annotations
We run the reference mapping using cell-level majority voting. You may adjust the k parameter to control the number of nearest neighbors to consider for voting.
[25] # Those functions are only used when faiss is not installed
def l2_sim(a, b):
sims = -np.linalg.norm(a - b, axis=1)
return sims
0s completed at 12:31 PM


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Haotian Cui
Ciliated
DC_activated
Dendritic cell
NK
NKT cells
pDC
To be predicted
Reference mapping and transfer the annotations
We run the reference mapping using cell-level majority voting. You may adjust the k parameter to control the number of nearest neighbors to consider for voting.
# Those functions are only used when faiss is not installed
def l2_sim(a, b):
sims = -np.linalg.norm(a - b, axis=1)
return sims
def get_similar_vectors(vector, ref, top_k=10):
# sims = cos_sim(vector, ref)
sims = l2_sim(vector, ref)
top_k_idx = np.argsort(sims)[::-1][:top_k]
return top_k_idx, sims[top_k_idx]
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
[26] ref_cell_embeddings = ref_embed_adata.obsm["X_scGPT"]
test_emebd = test_embed_adata.obsm["X_scGPT"]
k = 10 # number of neighbors
index = faiss.IndexFlatL2(ref_cell_embeddings.shape[1])
index.add(ref_cell_embeddings)
# Query dataset, k - number of closest elements (returns 2 numpy arrays)
0s completed at 12:31 PM


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
[25] sims = -np.linalg.norm(a - b, axis=1)
return sims
def get_similar_vectors(vector, ref, top_k=10):
# sims = cos_sim(vector, ref)
sims = l2_sim(vector, ref)
top_k_idx = np.argsort(sims)[::-1][:top_k]
return top_k_idx, sims[top_k_idx]
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: 'should_run_async' will not call 'transform_cell' automatically in the future. Please pass the result to 'transformed_cell' ar
and should_run_async(code)
ref_cell_embeddings = ref_embed_adata.obsm["X_scGPT"]
test_emebd = test_embed_adata.obsm["X_scGPT"]
k = 10 # number of neighbors
index = faiss.IndexFlatL2(ref_cell_embeddings.shape[1])
index.add(ref_cell_embeddings)
# Query dataset, k - number of closest elements (returns 2 numpy arrays)
distances, labels = index.search(test_emebd, k)
idx_list=[i for i in range(test_emebd.shape[0])]
preds = []
sim_list = distances
for k in idx_list:
if faiss_imported:
idx = labels[k]
else:
idx, sim = get_similar_vectors(test_emebd[k][np.newaxis, ...], ref_cell_embeddings, k)
pred = ref_embed_adata.obs[cell_type_key][idx].value_counts()
preds.append(pred.index[0])
gt = test_adata.obs[cell_type_key].to_numpy()
Evaluate the performance
[27] res_dict = {
0s completed at 12:31 PM
Haotian Cui



Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

Haotian Cui

[25]
```
        top_k_idx = np.argsort(sims)[::-1][:top_k]
        return top_k_idx, sims[top_k_idx]
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
  and should_run_async(code)
```

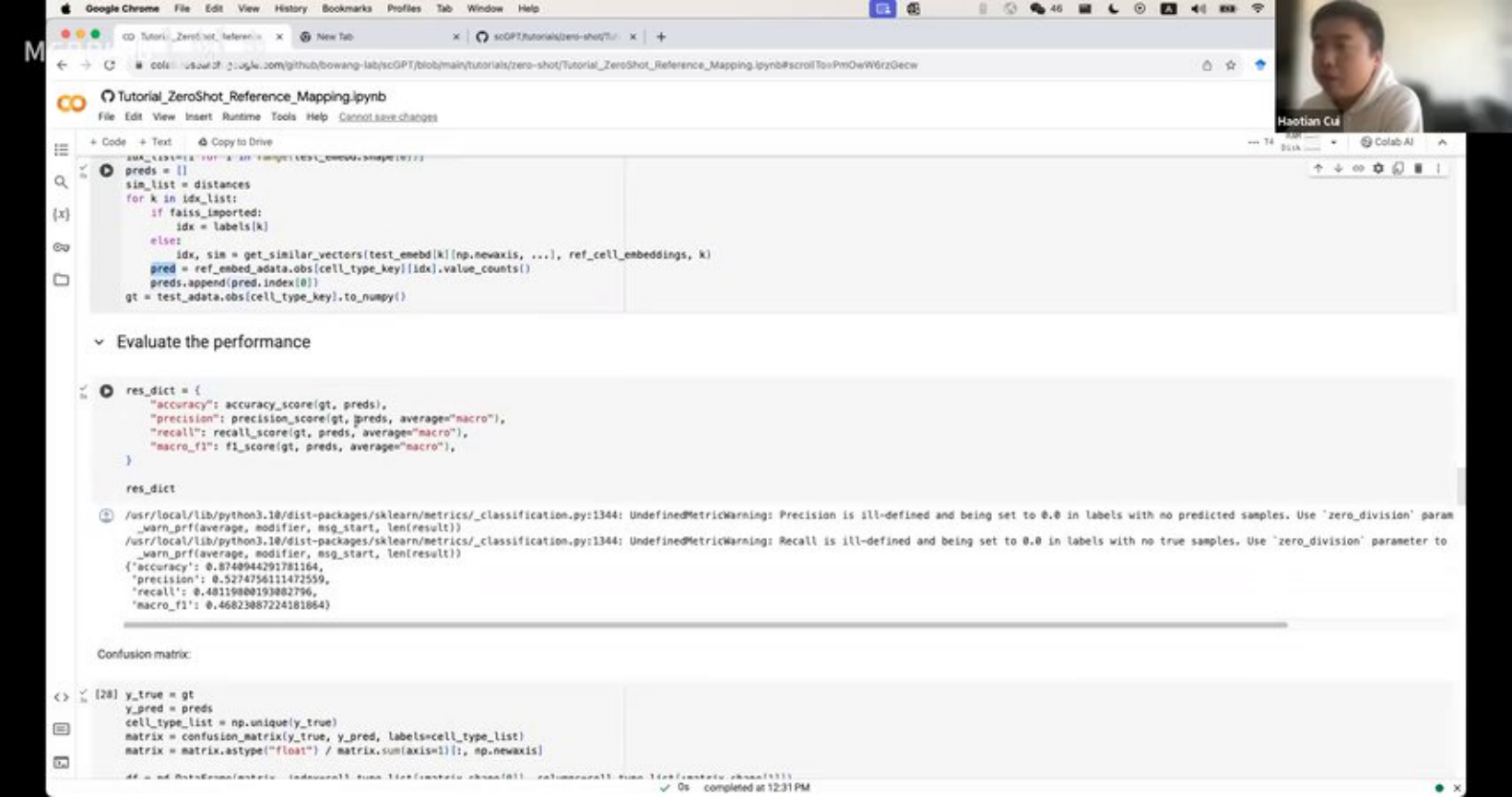
```
ref_cell_embeddings = ref_embed_adata.obsm["X_scGPT"]
test_emebd = test_embed_adata.obsm["X_scGPT"]

k = 10  # number of neighbors


index = faiss.IndexFlatL2(ref_cell_embeddings.shape[1])
index.add(ref_cell_embeddings)

# Query dataset, k - number of closest elements (returns 2 numpy arrays)
distances, labels = index.search(test_emebd, k)

idx_list=[i for i in range(test_emebd.shape[0])]
preds = []
sim_list = distances
for k in idx_list:
    if faiss_imported:
        idx = labels[k]
    else:
        idx, sim = get_similar_vectors(test_emebd[k][np.newaxis, ...], ref_cell_embeddings, k)
    pred = ref_embed_adata.obs[cell_type_key][idx].value_counts()
    preds.append(pred.index[0])
gt = test_adata.obs[cell_type_key].to_numpy()
```

Evaluate the performance

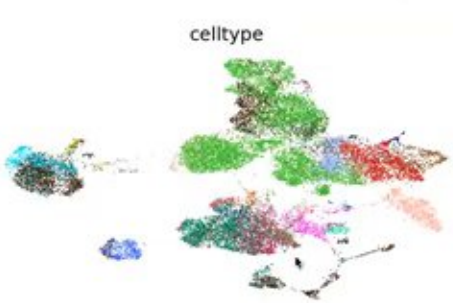
+ Code + Text

[27]
```
res_dict = {
    "accuracy": accuracy_score(gt, preds),
    "precision": precision_score(gt, preds, average="macro"),
    "recall": recall_score(gt, preds, average="macro"),
    "macro_f1": f1_score(gt, preds, average="macro"),
}

res_dict
```

0s completed at 12:31 PM


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
Haotian Cui
preds = []
sim_list = distances
for k in idx_list:
if faiss_imported:
idx = labels[k]
else:
idx, sim = get_similar_vectors(test_emebd[k][np.newaxis, ...], ref_cell_embeddings, k)
pred = ref_embed_adata.obs[cell_type_key][idx].value_counts()
preds.append(pred.index[0])
gt = test_adata.obs[cell_type_key].to_numpy()
Evaluate the performance
res_dict = {
"accuracy": accuracy_score(gt, preds),
"precision": precision_score(gt, preds, average="macro"),
"recall": recall_score(gt, preds, average="macro"),
"macro_f1": f1_score(gt, preds, average="macro"),
}
res_dict
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` param
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to
_warn_prf(average, modifier, msg_start, len(result))
{'accuracy': 0.8740944291781164,
'precision': 0.5274756111472559,
'recall': 0.4811980019302796,
'macro_f1': 0.4682308722418186}
Confusion matrix:
[28] y_true = gt
y_pred = preds
cell_type_list = np.unique(y_true)
matrix = confusion_matrix(y_true, y_pred, labels=cell_type_list)
matrix = matrix.astype("float") / matrix.sum(axis=1)[:, np.newaxis]
0s completed at 12:31 PM


Query
Reference


celltype
AT1
AT2
B cell
CCR7+ T
CD4+ T cells
CD8 T
CD8+ T cells
CD10+ B cells
CD14+ Monocytes
CD16+ Monocytes
CD20+ B cells
Ciliated
DC_activated
Dendritic cell
Erythrocytes
Erythroid progenitors
HSPCs
IGSF21+ Dendritic
M2 Macrophage
Macrophages
Mast cells
Megakaryocyte progenitors
Megakaryocytes
Monocyte progenitors
Monocytes
NK
NKT cells
Neutrophil
Plasma
Proliferating Macrophage
Proliferating T
Secretory
Signaling Alveolar Epithelial Type 2
T cell
Treg
Tregs
innate T
mDC
pDC
To be predicted

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
distances, labels = index.search(test_emebd, k)

idx_list=[i for i in range(test_emebd.shape[0])]
preds = []
sim_list = distances
for k in idx_list:
    if faiss_imported:
        idx = labels[k]
    else:
        idx, sim = get_similar_vectors(test_emebd[k][np.newaxis, ...], ref_cell_embeddings, k)
    pred = ref_embed_adata.obs[cell_type_key][idx].value_counts()
    preds.append(pred.index[0])
gt = test_adata.obs[cell_type_key].to_numpy()
```

Evaluate the performance

```
res_dict = {
    "accuracy": accuracy_score(gt, preds),
    "precision": precision_score(gt, preds, average="macro"),
    "recall": recall_score(gt, preds, average="macro"),
    "macro_f1": f1_score(gt, preds, average="macro"),
}

res_dict
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` param
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to
  _warn_prf(average, modifier, msg_start, len(result))
{'accuracy': 0.8740944291781164,
 'precision': 0.5274756111472559,
 'recall': 0.4811980019308279,
 'macro_f1': 0.4682308722418106}
```

Confusion matrix:

```
[28] y_true = gt
     y_pred = preds
     cell_type_list = np.unique(y_true)
     matrix = confusion_matrix(y_true, y_pred, labels=cell_type_list)
```

0s completed at 12:31 PM

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
        else:
            idx, sim = get_similar_vectors(test_emebd[k][np.newaxis, ...], ref_cell_embeddings, k)
        pred = ref_embed_adata.obs[cell_type_key][idx].value_counts()
        preds.append(pred.index[0])
    gt = test_adata.obs[cell_type_key].to_numpy()
```

Evaluate the performance

```
res_dict = {
    "accuracy": accuracy_score(gt, preds),
    "precision": precision_score(gt, preds, average="macro"),
    "recall": recall_score(gt, preds, average="macro"),
    "macro_f1": f1_score(gt, preds, average="macro"),
}

res_dict
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` param
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to
  _warn_prf(average, modifier, msg_start, len(result))
{'accuracy': 0.8740944291781164,
 'precision': 0.5274756111472559,
 'recall': 0.4811980019302796,
 'macro_f1': 0.4682308722418186}
```

Confusion matrix:

```
[28] y_true = gt
     y_pred = preds
     cell_type_list = np.unique(y_true)
     matrix = confusion_matrix(y_true, y_pred, labels=cell_type_list)
     matrix = matrix.astype("float") / matrix.sum(axis=1)[:, np.newaxis]

     df = pd.DataFrame(matrix, index=cell_type_list[:matrix.shape[0]], columns=cell_type_list[:matrix.shape[1]])

     ax = sns.clustermap(df,
                         cmap='Purples',
                         annot=True ,fmt=".2f",
                         annot_kws={'size': 8},
```

0s completed at 12:31 PM

Haotian Cui

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

Colab AI

Evaluate the performance

```
[27] res_dict = {
        "accuracy": accuracy_score(gt, preds),
        "precision": precision_score(gt, preds, average="macro"),
        "recall": recall_score(gt, preds, average="macro"),
        "macro_f1": f1_score(gt, preds, average="macro"),
    }

    res_dict
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` param
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to
  _warn_prf(average, modifier, msg_start, len(result))
{'accuracy': 0.8740944291781164,
 'precision': 0.5274756111472559,
 'recall': 0.4811980019302796,
 'macro_f1': 0.4682308722418106}
```

Confusion matrix:

```
y_true = gt
y_pred = preds
cell_type_list = np.unique(y_true)
matrix = confusion_matrix(y_true, y_pred, labels=cell_type_list)
matrix = matrix.astype("float") / matrix.sum(axis=1)[:, np.newaxis]

df = pd.DataFrame(matrix, index=cell_type_list[:matrix.shape[0]], columns=cell_type_list[:matrix.shape[1]])

ax = sns.clustermap(df,
                    cmap='Purples',
                    annot=True ,fmt=".2f",
                    annot_kws={'size': 8},
                    vmin=0,
                    vmax=1,
                    row_cluster=False,
                    col_cluster=False,
                    figsize=(14, 14))
```

0s completed at 12:31 PM

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
y_true = gt
y_pred = preds
cell_type_list = np.unique(y_true)
matrix = confusion_matrix(y_true, y_pred, labels=cell_type_list)
matrix = matrix.astype("float") / matrix.sum(axis=1)[:, np.newaxis]

df = pd.DataFrame(matrix, index=cell_type_list[:matrix.shape[0]], columns=cell_type_list[:matrix.shape[1]])

ax = sns.clustermap(df,
                    cmap='Purples',
                    annot=True ,fmt=".2f",
                    annot_kws={'size': 8},
                    vmin=0,
                    vmax=1,
                    row_cluster=False,
                    col_cluster=False,
                    figsize=(14, 14))
```

0s completed at 12:31 PM

Haotian Cui

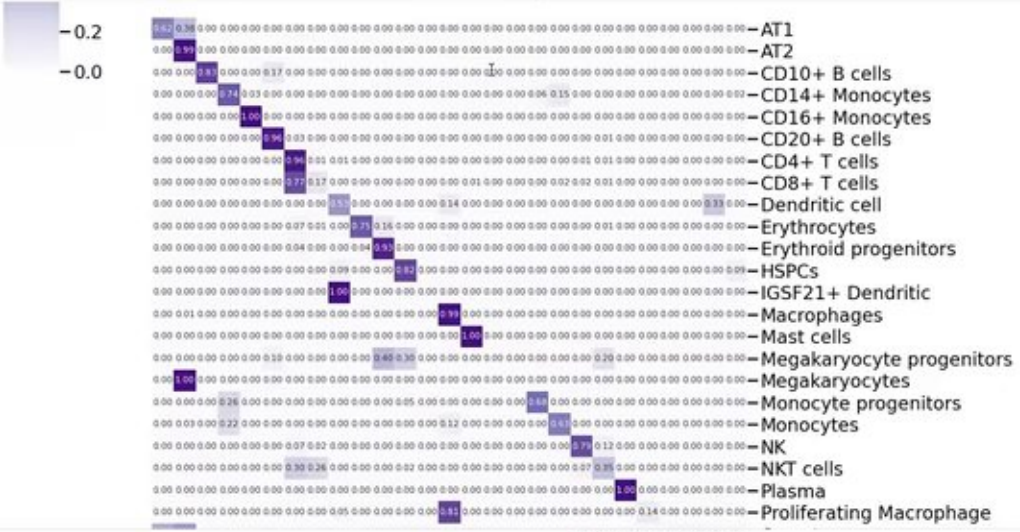
Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Tutorial_ZeroShot_Reference_Mapping.ipynb

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

```
ax = sns.clustermap(df,
                    cmap='Purples',
                    annot=True ,fmt=".2f",
                    annot_kws={'size': 8},
                    vmin=0,
                    vmax=1,
                    row_cluster=False,
                    col_cluster=False,
                    figsize=(14, 14))
```



0s completed at 12:31 PM

Haotian Cui

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

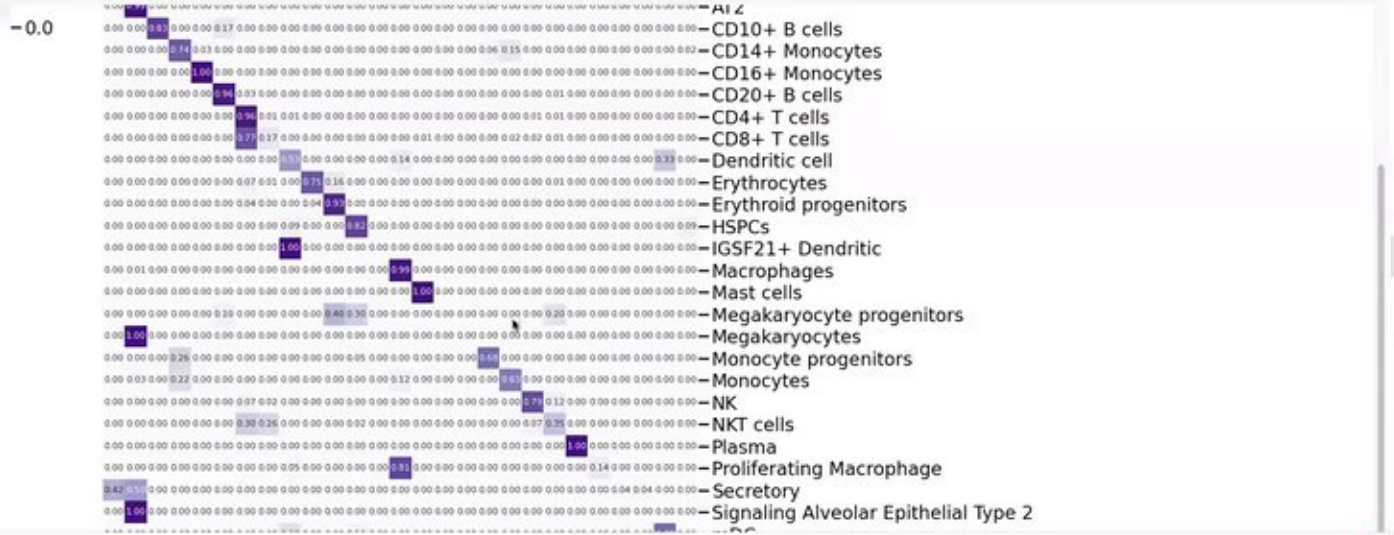
Haotian Cui

Tutorial_ZeroShot_Reference_Mapping.ipynb

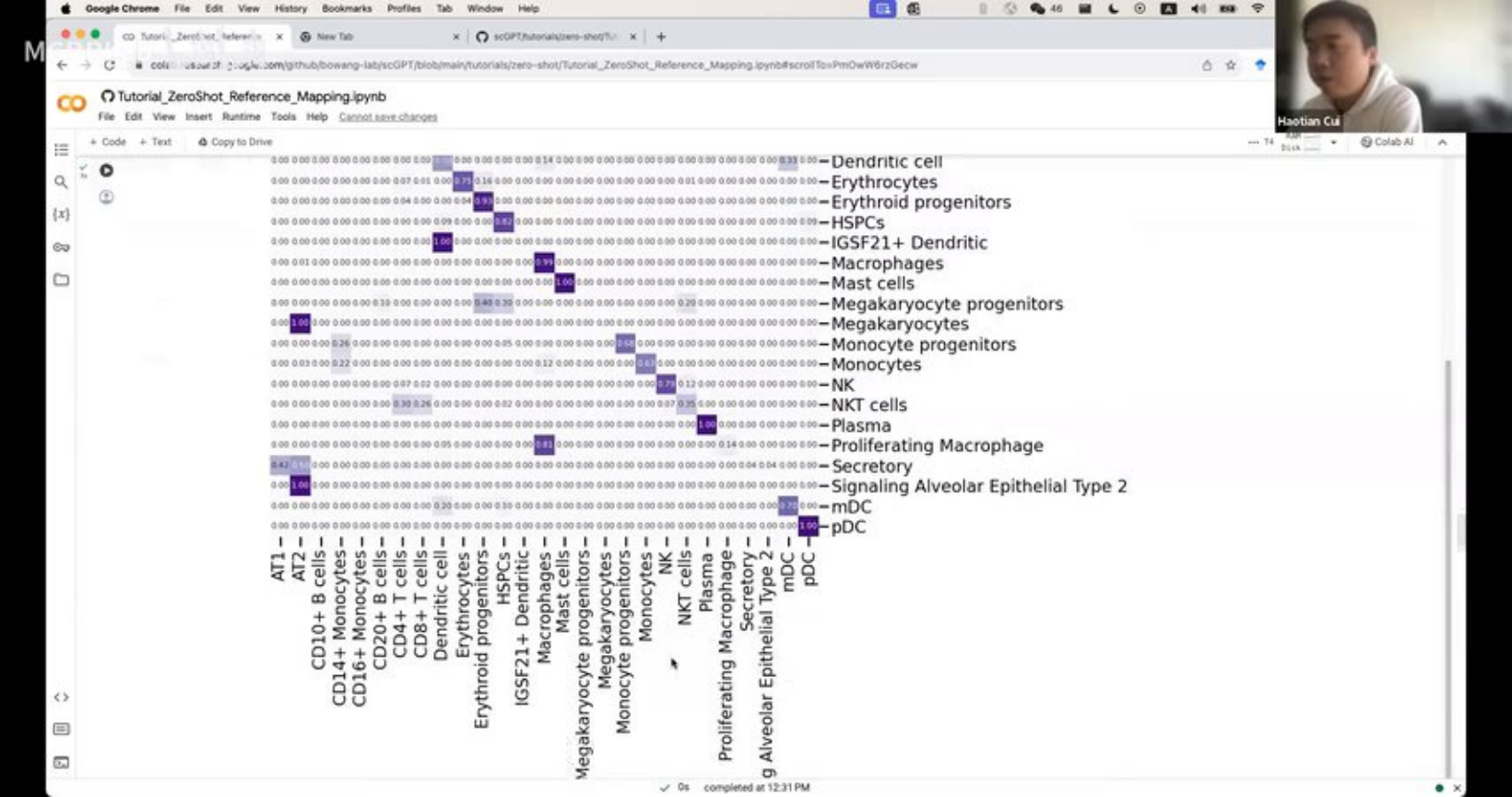
File Edit View Insert Runtime Tools Help Cannot save changes

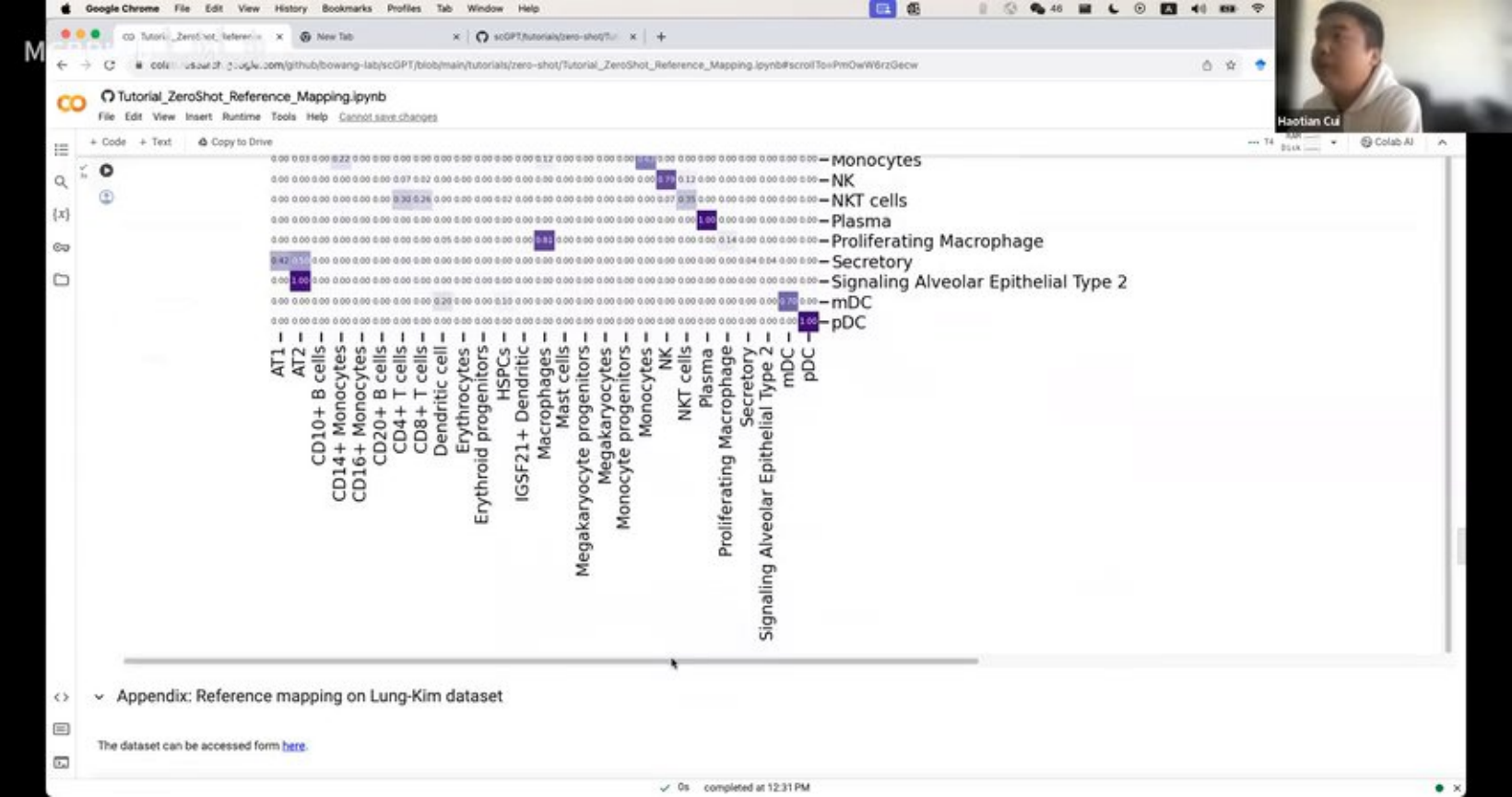
+ Code + Text Copy to Drive

```
ax = sns.clustermap(df,
                    cmap='Purples',
                    annot=True ,fmt=".2f",
                    annot_kws={'size': 8},
                    vmin=0,
                    vmax=1,
                    row_cluster=False,
                    col_cluster=False,
                    figsize=(14, 14))
```



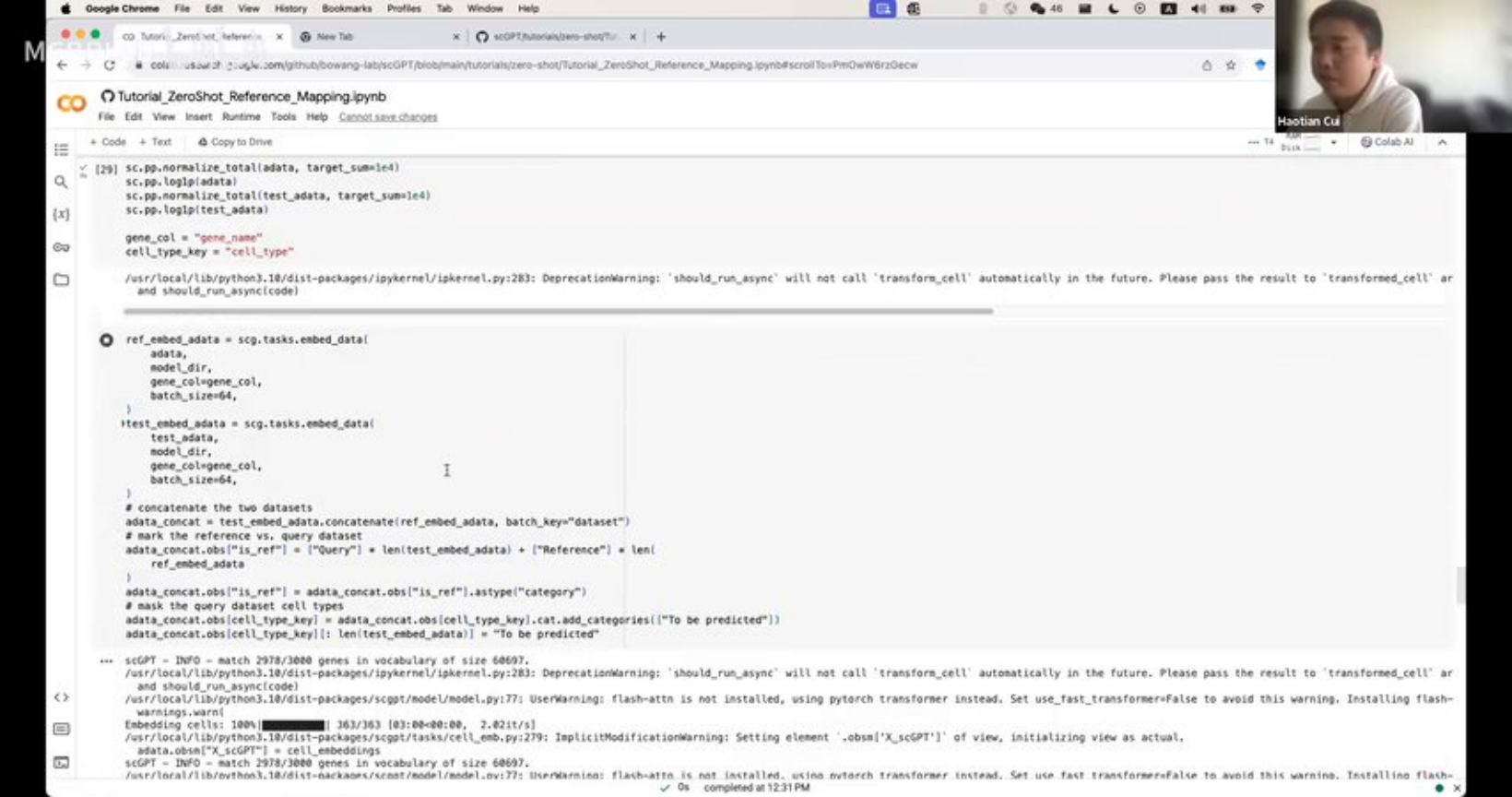
0s completed at 12:31 PM

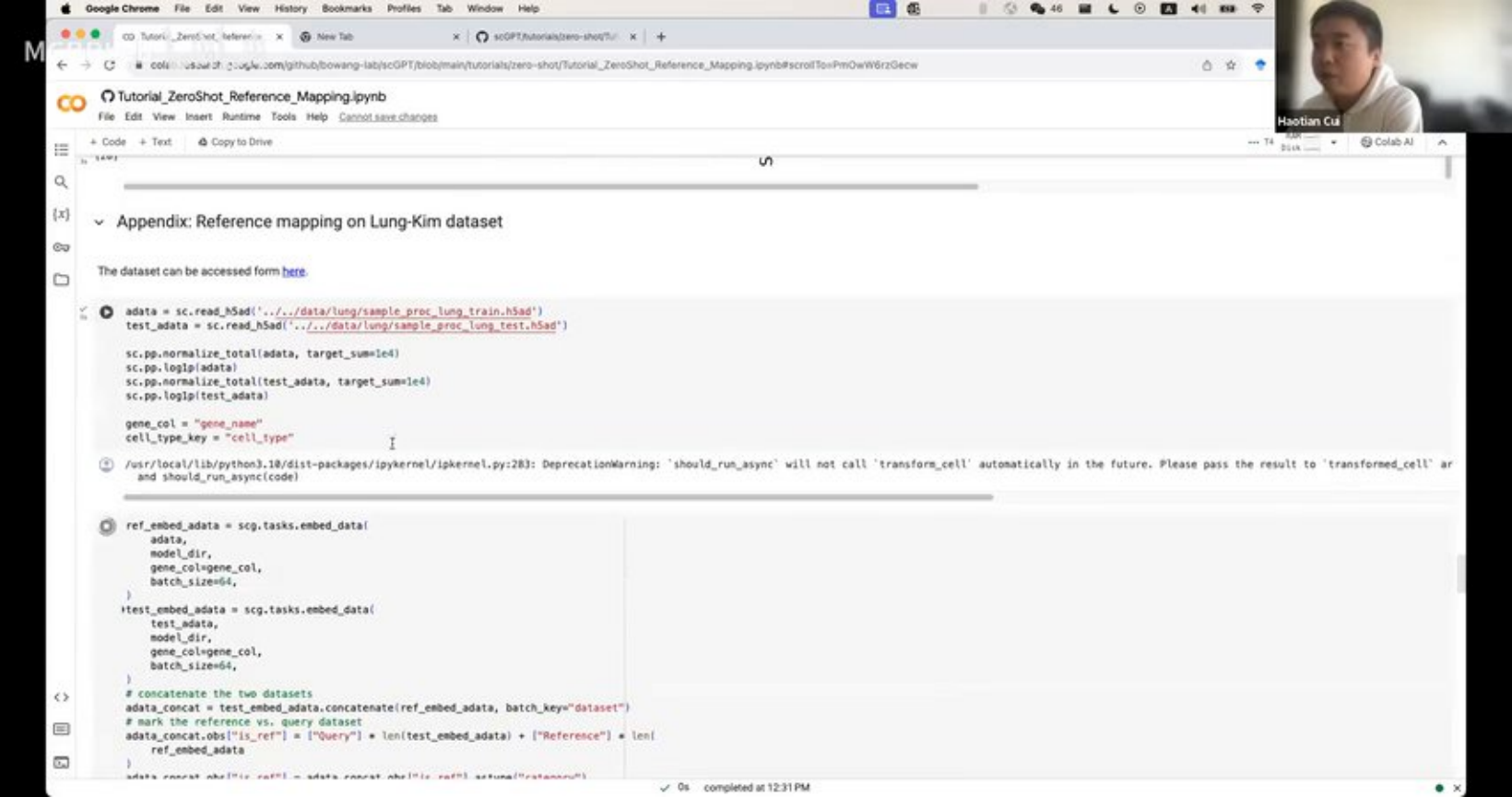

Tutorial_ZeroShot_Reference_Mapping.ipynb
Dendritic cell
Erythrocytes
Erythroid progenitors
HSPCs
IGSF21+ Dendritic
Macrophages
Mast cells
Megakaryocyte progenitors
Megakaryocytes
Monocyte progenitors
Monocytes
NK
NKT cells
Plasma
Proliferating Macrophage
Secretory
Signaling Alveolar Epithelial Type 2
mDC
pDC
AT1
AT2
CD10+ B cells
CD14+ Monocytes
CD16+ Monocytes
CD20+ B cells
CD4+ T cells
CD8+ T cells
Dendritic cell
Erythrocytes
Erythroid progenitors
HSPCs
IGSF21+ Dendritic
Macrophages
Mast cells
Megakaryocyte progenitors
Megakaryocytes
Monocyte progenitors
Monocytes
NK
NKT cells
Plasma
Proliferating Macrophage
Secretory
g Alveolar Epithelial Type 2
mDC
pDC
completed at 12:31 PM

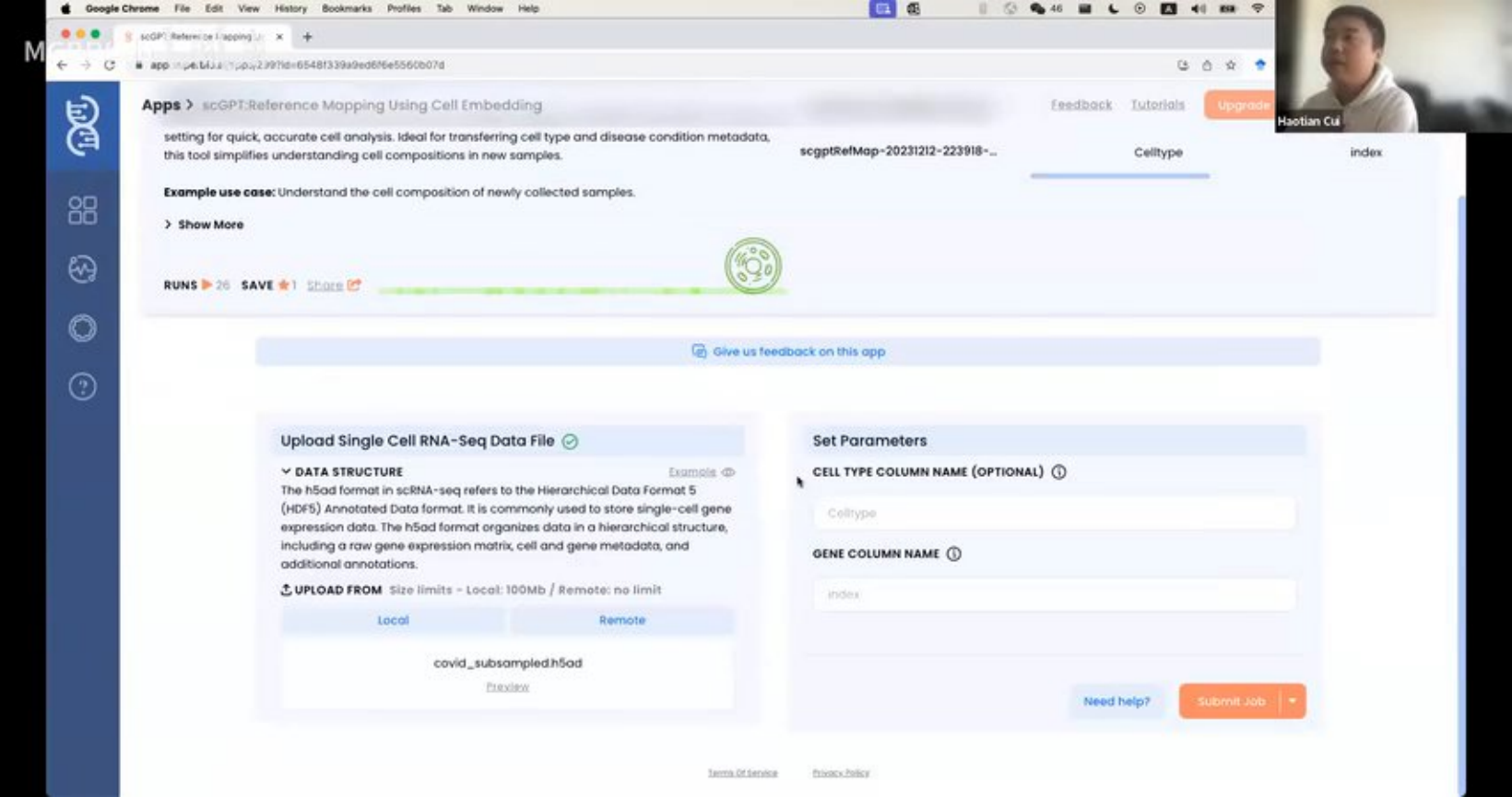


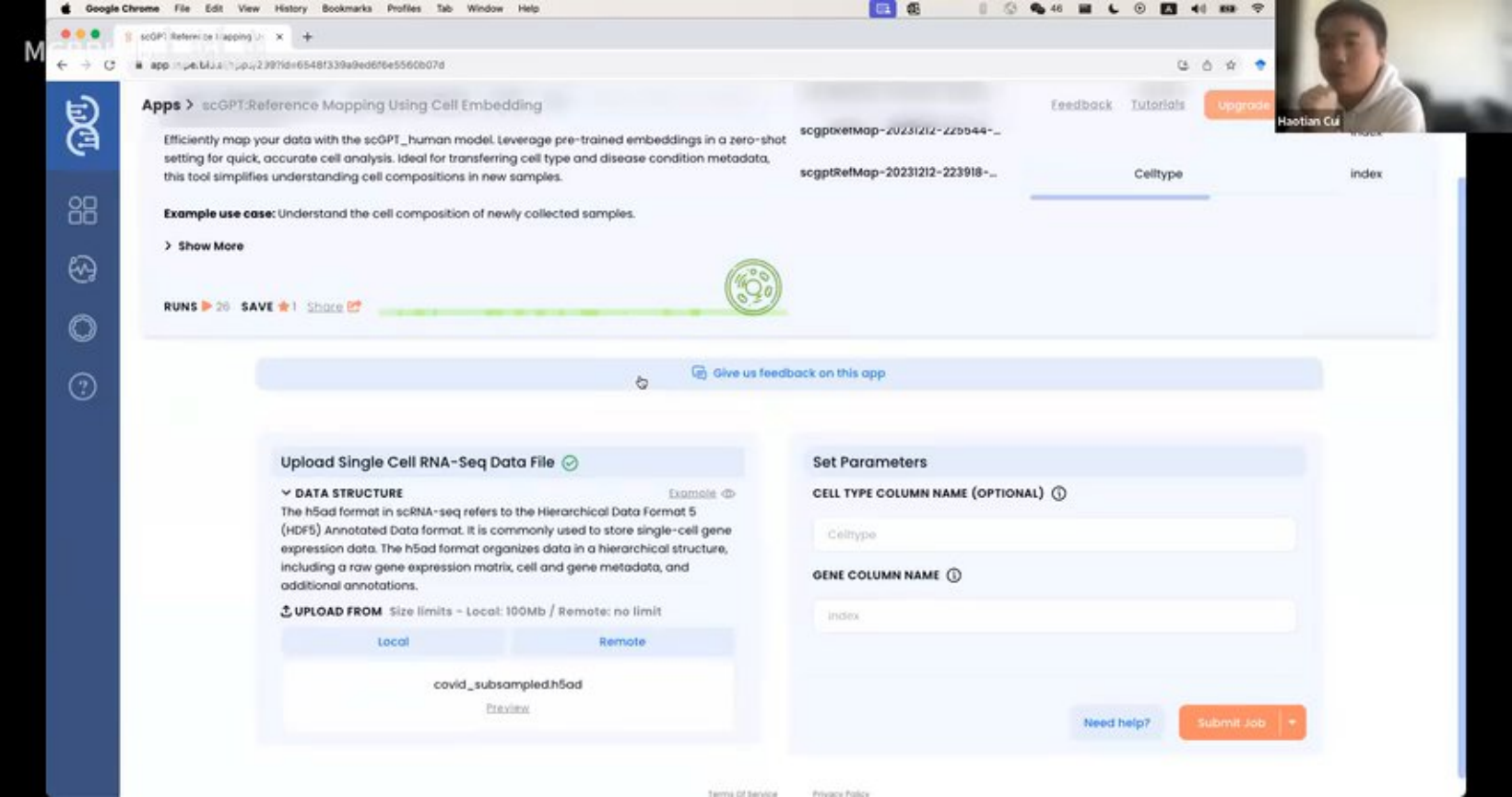
Appendix: Reference mapping on Lung-Kim dataset

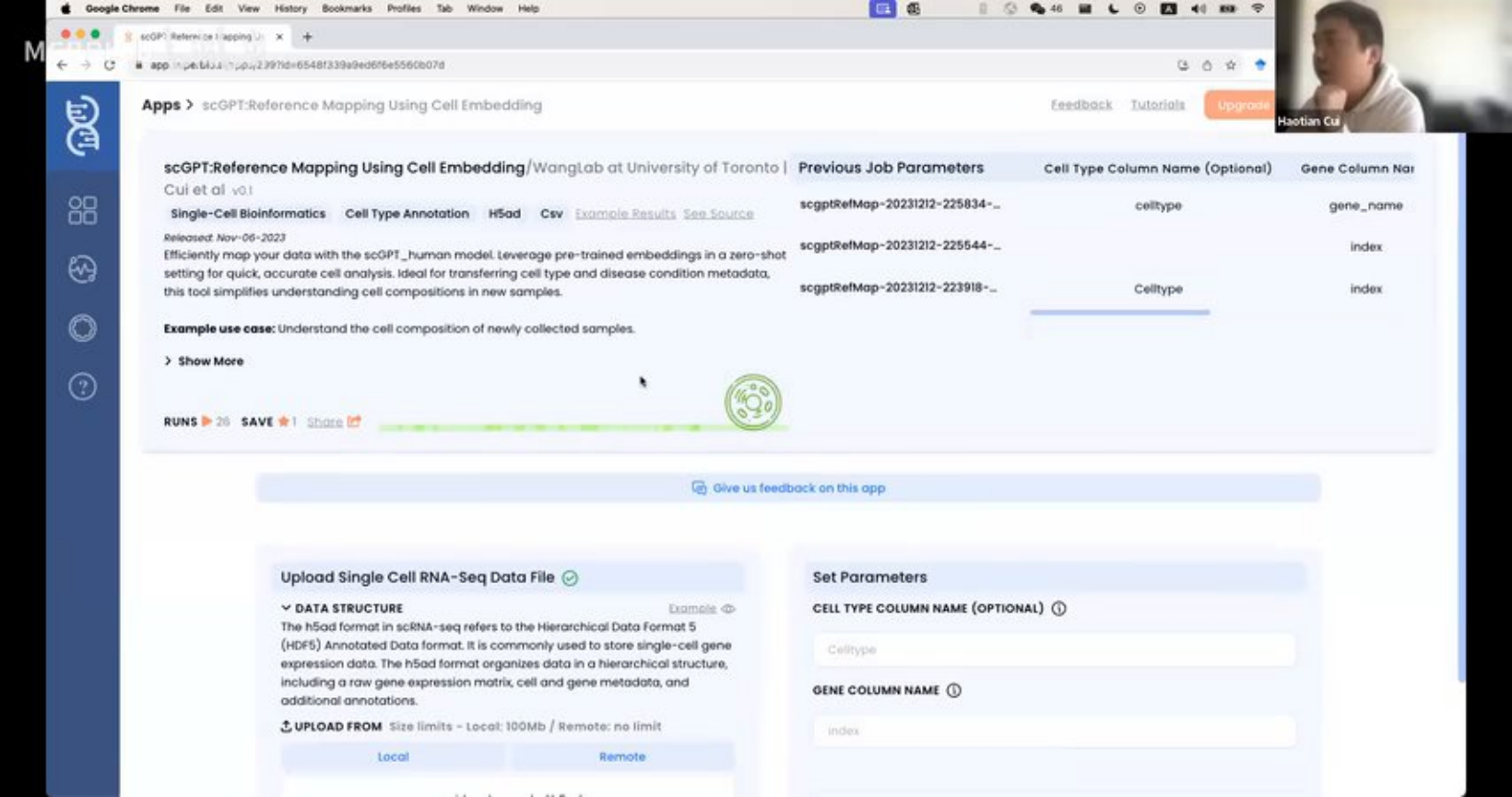
The dataset can be accessed form here.


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
[29] sc.pp.normalize_total(adata, target_sum=1e4)
sc.pp.log1p(adata)
sc.pp.normalize_total(test_adata, target_sum=1e4)
sc.pp.log1p(test_adata)
gene_col = "gene_name"
cell_type_key = "cell_type"
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: 'should_run_async' will not call 'transform_cell' automatically in the future. Please pass the result to 'transformed_cell' ar
and should_run_async(code)
ref_embed_adata = scg.tasks.embed_data(
adata,
model_dir,
gene_col=gene_col,
batch_size=64,
)
test_embed_adata = scg.tasks.embed_data(
test_adata,
model_dir,
gene_col=gene_col,
batch_size=64,
)
# concatenate the two datasets
adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
# mark the reference vs. query dataset
adata_concat.obs["is_ref"] = ["Query"] * len(test_embed_adata) + ["Reference"] * len(
ref_embed_adata
)
adata_concat.obs["is_ref"] = adata_concat.obs["is_ref"].astype("category")
# mask the query dataset cell types
adata_concat.obs[cell_type_key] = adata_concat.obs[cell_type_key].cat.add_categories(["To be predicted"])
adata_concat.obs[cell_type_key][: len(test_embed_adata)] = "To be predicted"
scGPT - INFO - match 2978/3000 genes in vocabulary of size 60697.
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: 'should_run_async' will not call 'transform_cell' automatically in the future. Please pass the result to 'transformed_cell' ar
and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/scgpt/model/model.py:77: UserWarning: flash-attn is not installed, using pytorch transformer instead. Set use_fast_transformer=False to avoid this warning. Installing flash-
warnings.warn(
Embedding cells: 100%| | 363/363 [03:00<00:00, 2.02it/s]
/usr/local/lib/python3.10/dist-packages/scgpt/tasks/cell_emb.py:279: ImplicitModificationWarning: Setting element `.obsm['X_scGPT']` of view, initializing view as actual.
adata.obsm["X_scGPT"] = cell_embeddings
scGPT - INFO - match 2978/3000 genes in vocabulary of size 60697.
0s completed at 12:31 PM
Haotian Cui


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tutorial_ZeroShot_Reference_Mapping.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Colab AI
Haotian Cui
Appendix: Reference mapping on Lung-Kim dataset
The dataset can be accessed form here.
adata = sc.read_h5ad('../../data/lung/sample_proc_lung_train.h5ad')
test_adata = sc.read_h5ad('../../data/lung/sample_proc_lung_test.h5ad')
sc.pp.normalize_total(adata, target_sum=1e4)
sc.pp.log1p(adata)
sc.pp.normalize_total(test_adata, target_sum=1e4)
sc.pp.log1p(test_adata)
gene_col = "gene_name"
cell_type_key = "cell_type"
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pass the result to `transformed_cell` ar
and should_run_async(code)
ref_embed_adata = scg.tasks.embed_data(
adata,
model_dir,
gene_col=gene_col,
batch_size=64,
)
test_embed_adata = scg.tasks.embed_data(
test_adata,
model_dir,
gene_col=gene_col,
batch_size=64,
)
# concatenate the two datasets
adata_concat = test_embed_adata.concatenate(ref_embed_adata, batch_key="dataset")
# mark the reference vs. query dataset
adata_concat.obs["is_ref"] = ["Query"] * len(test_embed_adata) + ["Reference"] * len(
ref_embed_adata
)
0s completed at 12:31 PM


Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
scgptRefMap-20231212-223918-...
Celltype
index
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26 SAVE 1 Share
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index
Need help?
Submit Job
Terms Of Service
Privacy Policy


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26 SAVE 1 Share
scgptRefMap-20231212-223918-...
Celltype
index
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index
Need help?
Submit Job


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
scGPT:Reference Mapping Using Cell Embedding/WangLab at University of Toronto | Cui et al v0.1
Single-Cell Bioinformatics
Cell Type Annotation
H5ad
Csv
Example Results
See Source
Released: Nov-06-2023
Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26
SAVE 1
Share
Previous Job Parameters
Cell Type Column Name (Optional)
Gene Column Nar
scgptRefMap-20231212-225834-...
celltype
gene_name
scgptRefMap-20231212-225544-...
index
scgptRefMap-20231212-223918-...
Celltype
index
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help

Haotian Cui

Apps > scGPT:Reference Mapping Using Cell Embedding

Feedback Tutorials Upgrade

Released: Nov-06-2023

Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.

Example use case: Understand the cell composition of newly collected samples.

> Show More

RUNS ▶ 26 SAVE ★ 1 Share

scgptRefMap-20231212-225544-... index

scgptRefMap-20231212-223918-... Celltype index

Give us feedback on this app

Upload Single Cell RNA-Seq Data File

DATA STRUCTURE Example

The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.

UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit

Local Remote

covid_subsampled.h5ad

Preview

Set Parameters

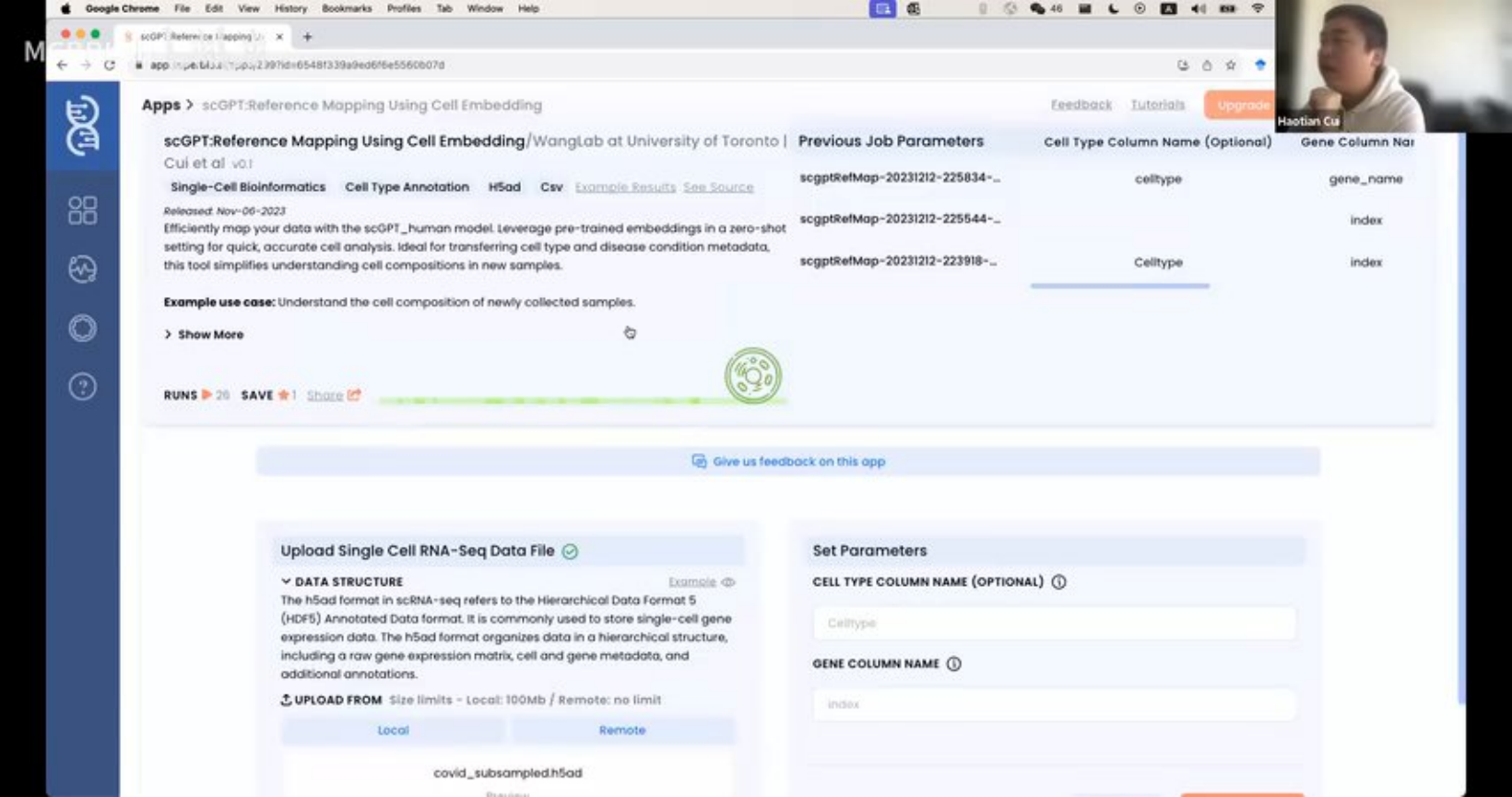
CELL TYPE COLUMN NAME (OPTIONAL)

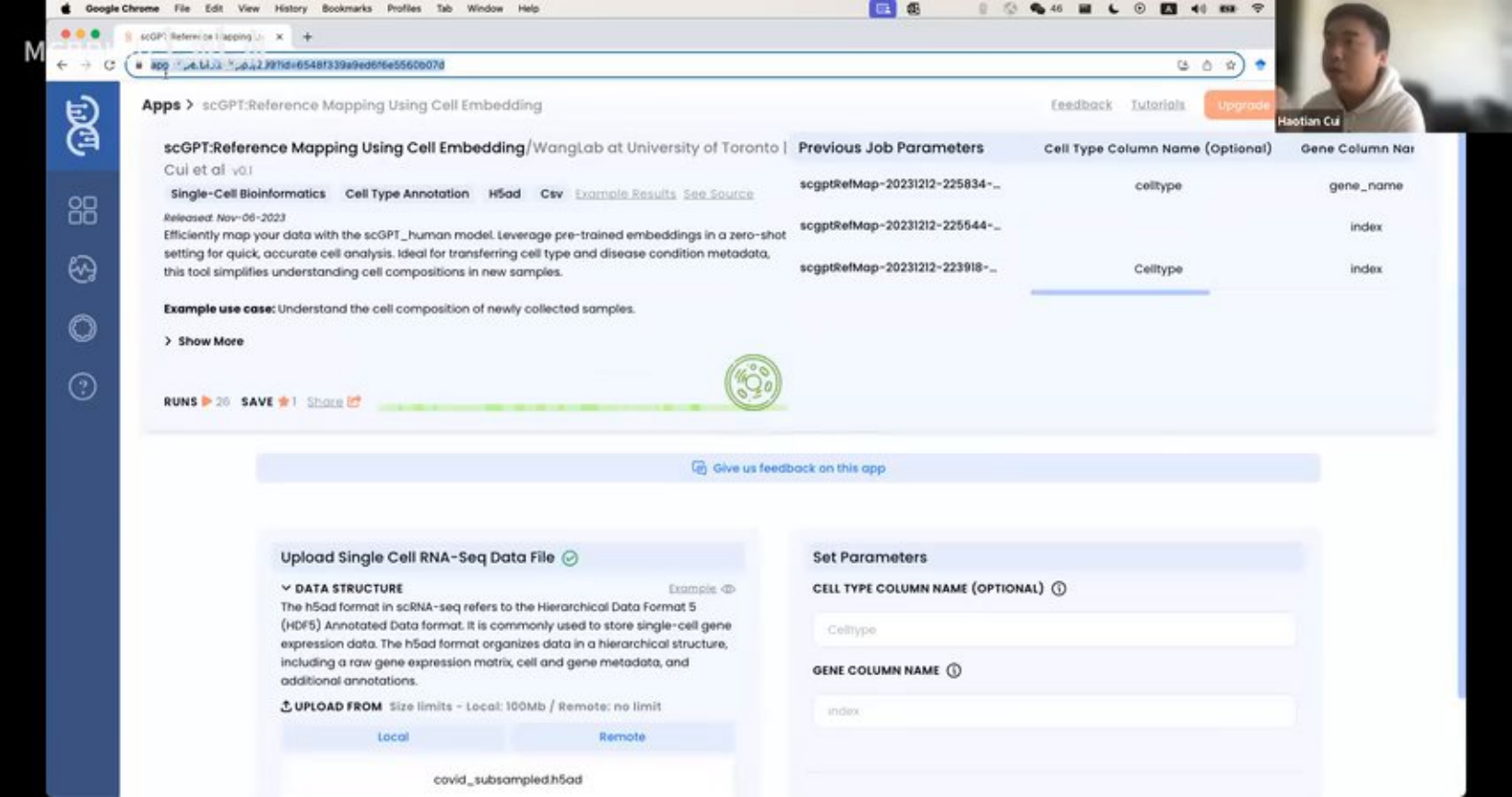
Celltype

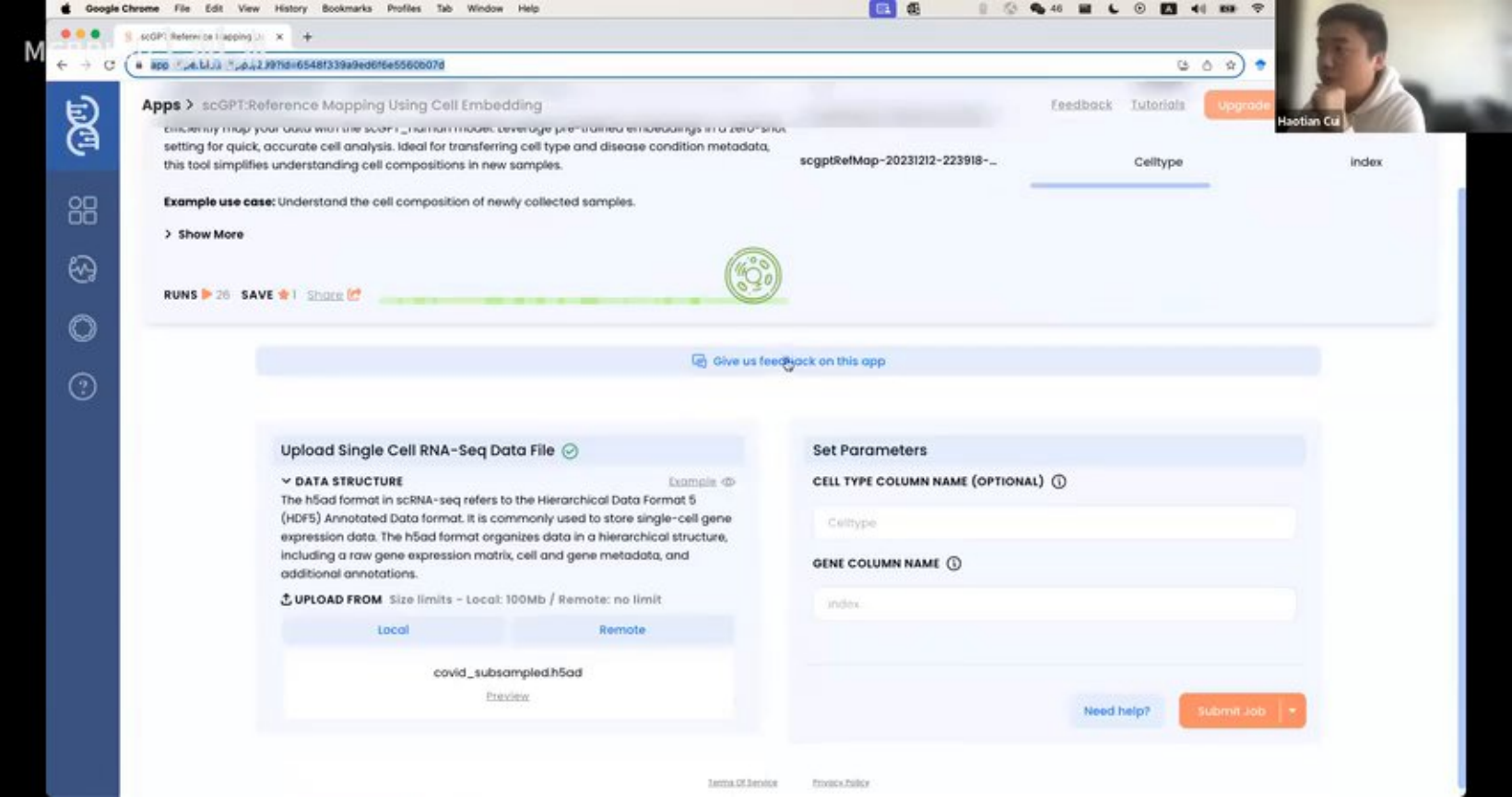
GENE COLUMN NAME

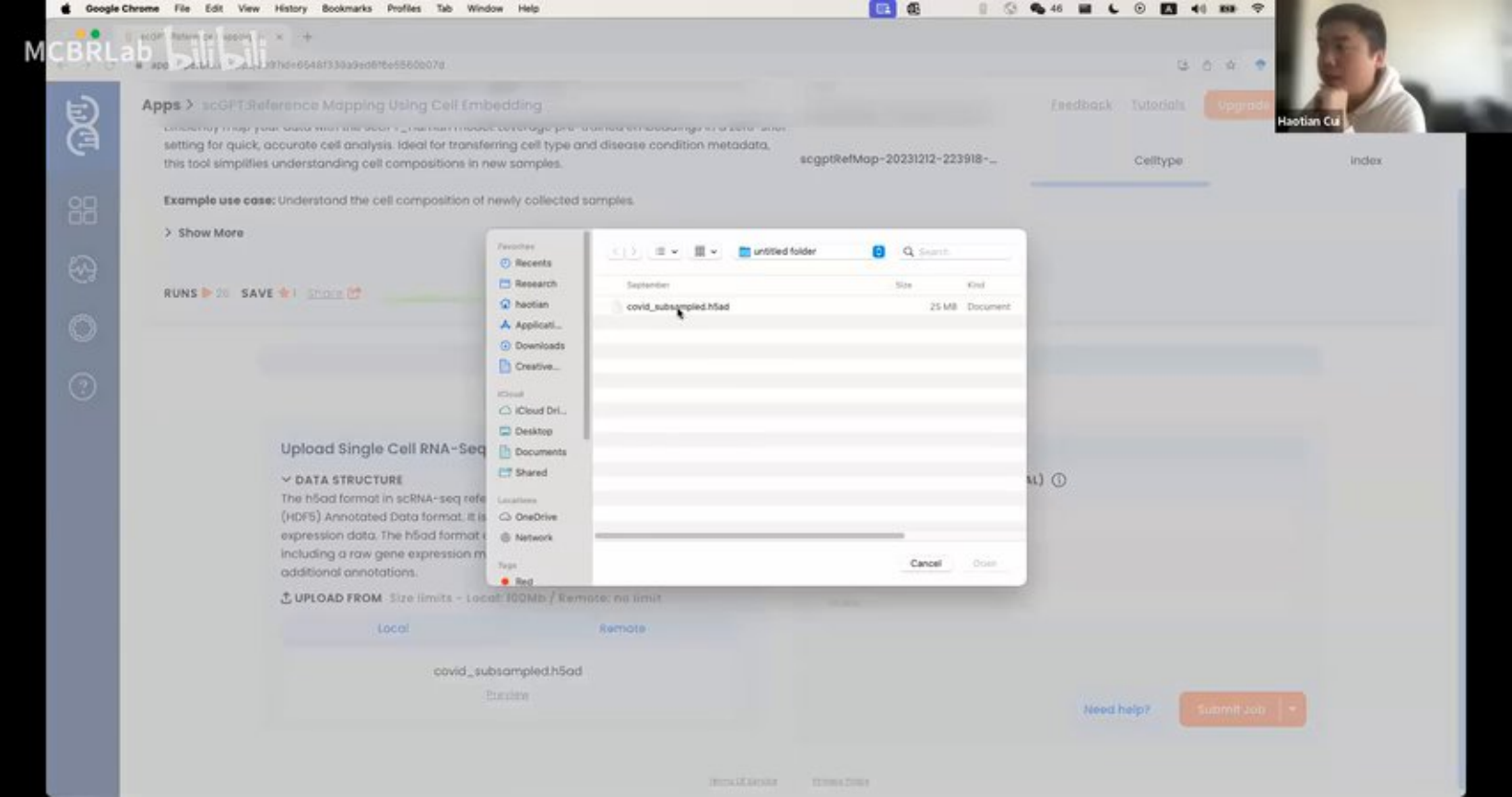
index

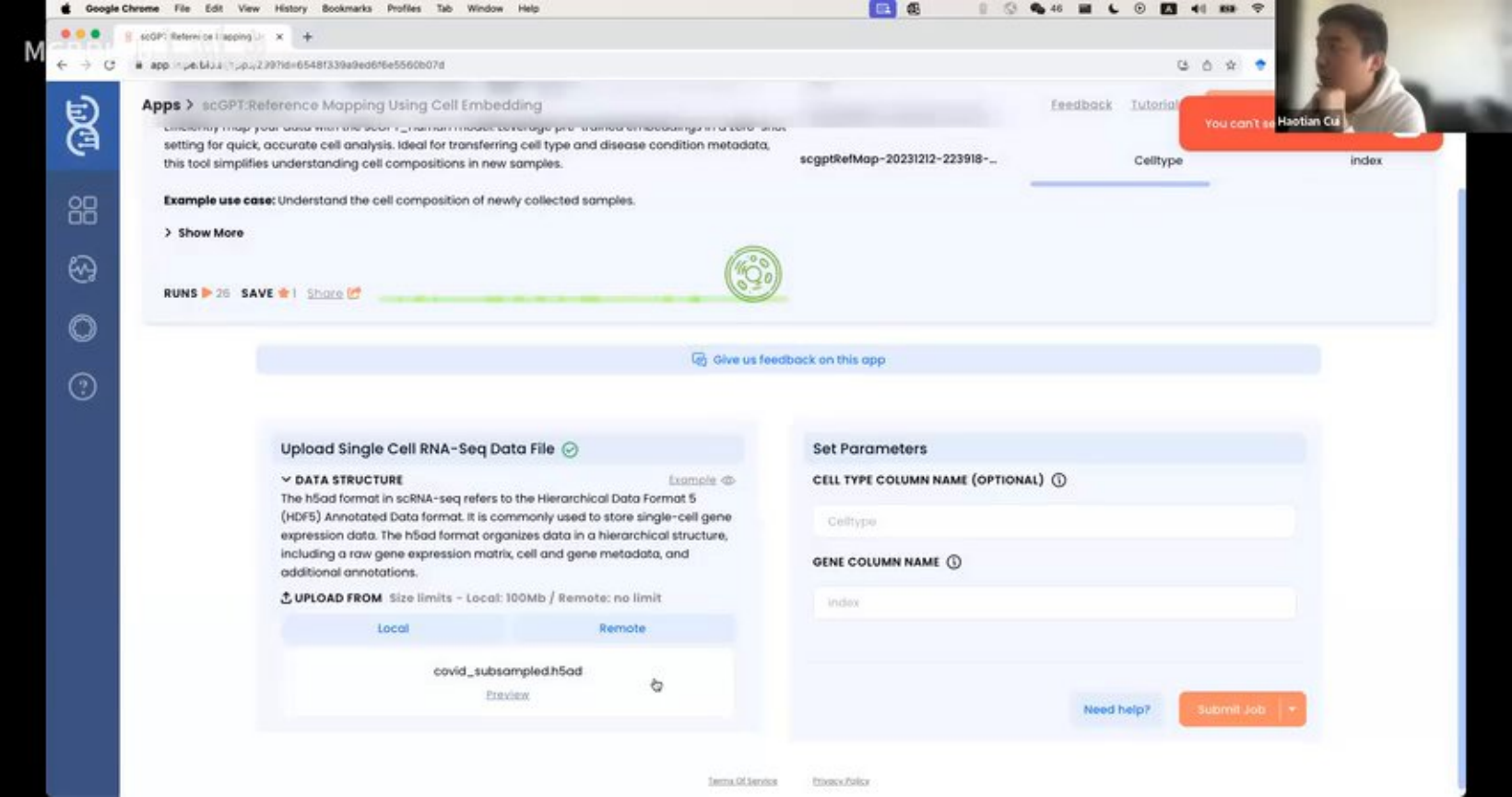
Need help? Submit Job

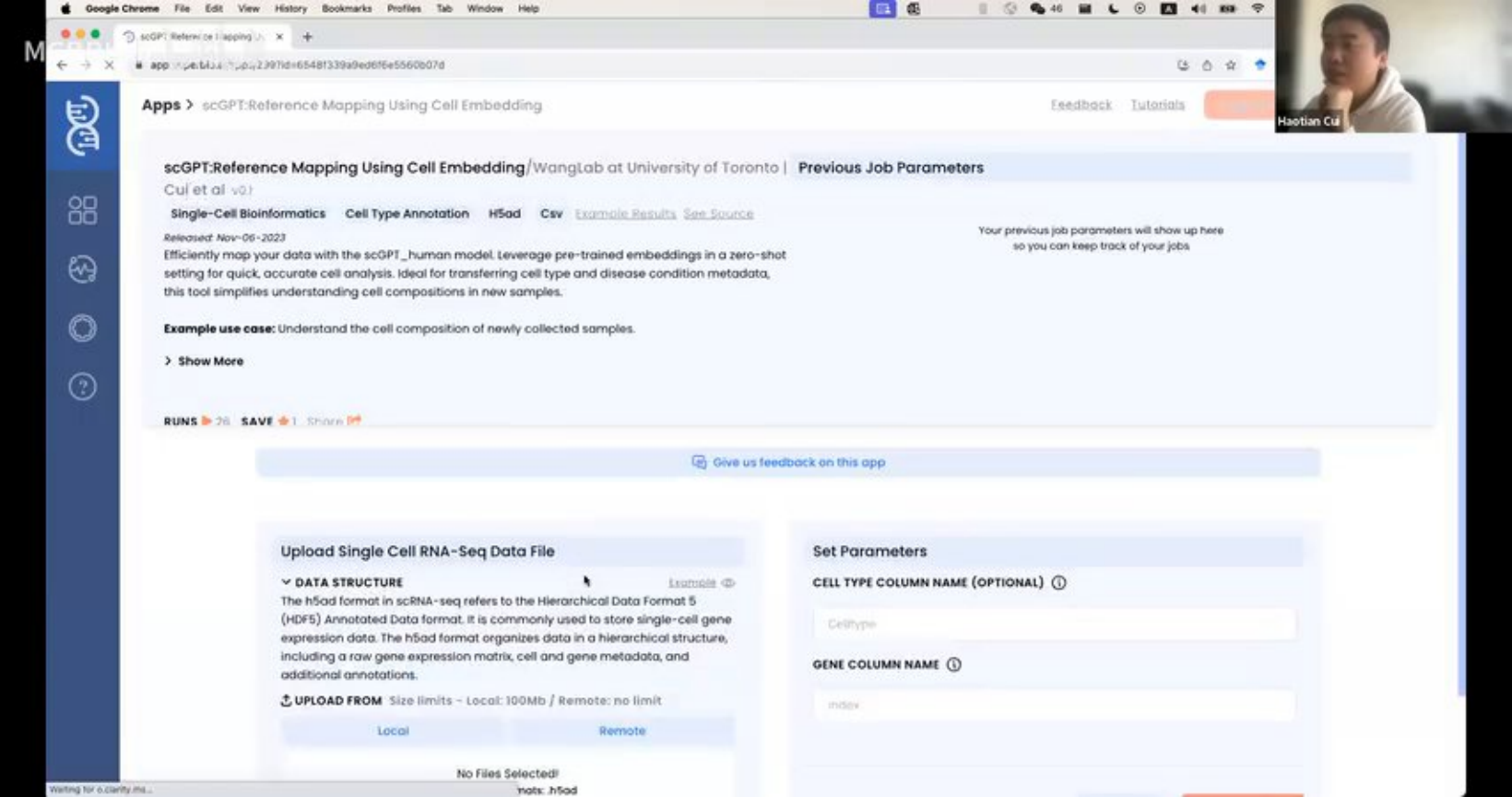

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback Tutorials Upgrade
scGPT:Reference Mapping Using Cell Embedding/WangLab at University of Toronto | Cui et al v0.1
Single-Cell Bioinformatics Cell Type Annotation H5ad Csv Example Results See Source
Released: Nov-06-2023
Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
Example use case: Understand the cell composition of newly collected samples.
> Show More
RUNS 26 SAVE 1 Share
Previous Job Parameters
Cell Type Column Name (Optional)
Gene Column Nar
scgptRefMap-20231212-225834-... celltype gene_name
scgptRefMap-20231212-225544-... index
scgptRefMap-20231212-223918-... Celltype index
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local Remote
covid_subsampled.h5ad
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
scGPT:Reference Mapping Using Cell Embedding/WangLab at University of Toronto | Cui et al v0.1
Single-Cell Bioinformatics
Cell Type Annotation
H5ad
Csv
Example Results
See Source
Released: Nov-06-2023
Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26
SAVE 1
Share
Previous Job Parameters
Cell Type Column Name (Optional)
Gene Column Nar
scgptRefMap-20231212-225834-...
celltype
gene_name
scgptRefMap-20231212-225544-...
index
scgptRefMap-20231212-223918-...
Celltype
index
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
scgptRefMap-20231212-223918-...
Celltype
index
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26 SAVE 1 Share
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index
Need help?
Submit Job
Terms Of Service
Privacy Policy


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
MCBRLab
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata,
this tool simplifies understanding cell compositions in new samples.
scgptRefMap-20231212-223918-...
Celltype
index
Example use case: Understand the cell composition of newly collected samples.
> Show More
RUNS 26 SAVE 1 Share
untitled folder
Recents
Research
haotian
Applicati...
Downloads
Creative...
iCloud Dri...
Desktop
Documents
Shared
OneDrive
Network
Red
September
Size
Kind
covid_subsampled.h5ad
25 MB
Document
Cancel
Upload Single Cell RNA-Seq
DATA STRUCTURE
The h5ad format in scRNA-seq refe
(HDF5) Annotated Data format. It is
expression data. The h5ad format
including a raw gene expression m
additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Need help?
Submit Job


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorial
setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
scgptRefMap-20231212-223918-...
Celltype
index
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26 SAVE Share
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index
Need help?
Submit Job
Terms Of Service
Privacy Policy


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
scGPT:Reference Mapping Using Cell Embedding/WangLab at University of Toronto | Cui et al v0.1
Previous Job Parameters
Single-Cell Bioinformatics
Cell Type Annotation
H5ad
Csv
Example Results
See Source
Released: Nov-06-2023
Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
Example use case: Understand the cell composition of newly collected samples.
> Show More
Your previous job parameters will show up here so you can keep track of your jobs
RUNS 26 SAVE 1 Share
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
No Files Selected!
Formats: h5ad
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index

Apps > scGPT:Reference Mapping Using Cell Embedding

Feedback Tutorials Upgrade

Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.

Example use case: Understand the cell composition of newly collected samples.

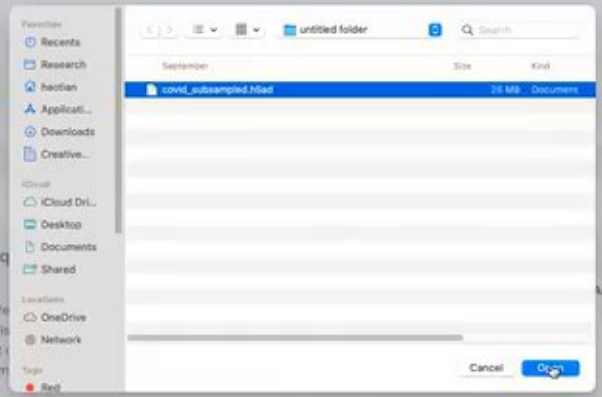
> Show More

RUNS 76 SAVE 1 Share

scgptRefMap-20231212-223918-...

Celltype

index



Upload Single Cell RNA-Seq

DATA STRUCTURE

The h5ad format in scRNA-seq refe... (HDF5) Annotated Data format. It is... expression data. The h5ad format ... including a raw gene expression m... additional annotations.

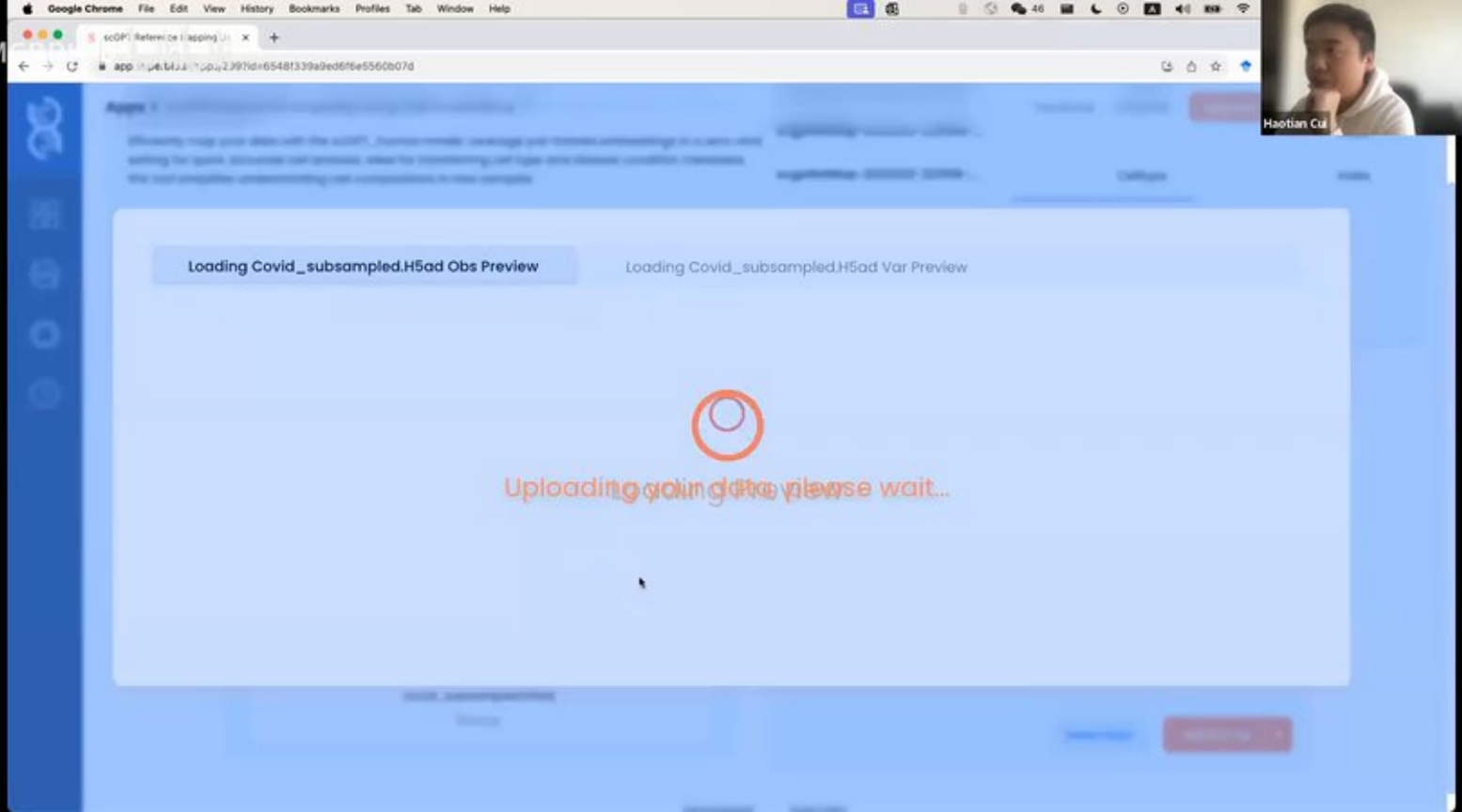
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit

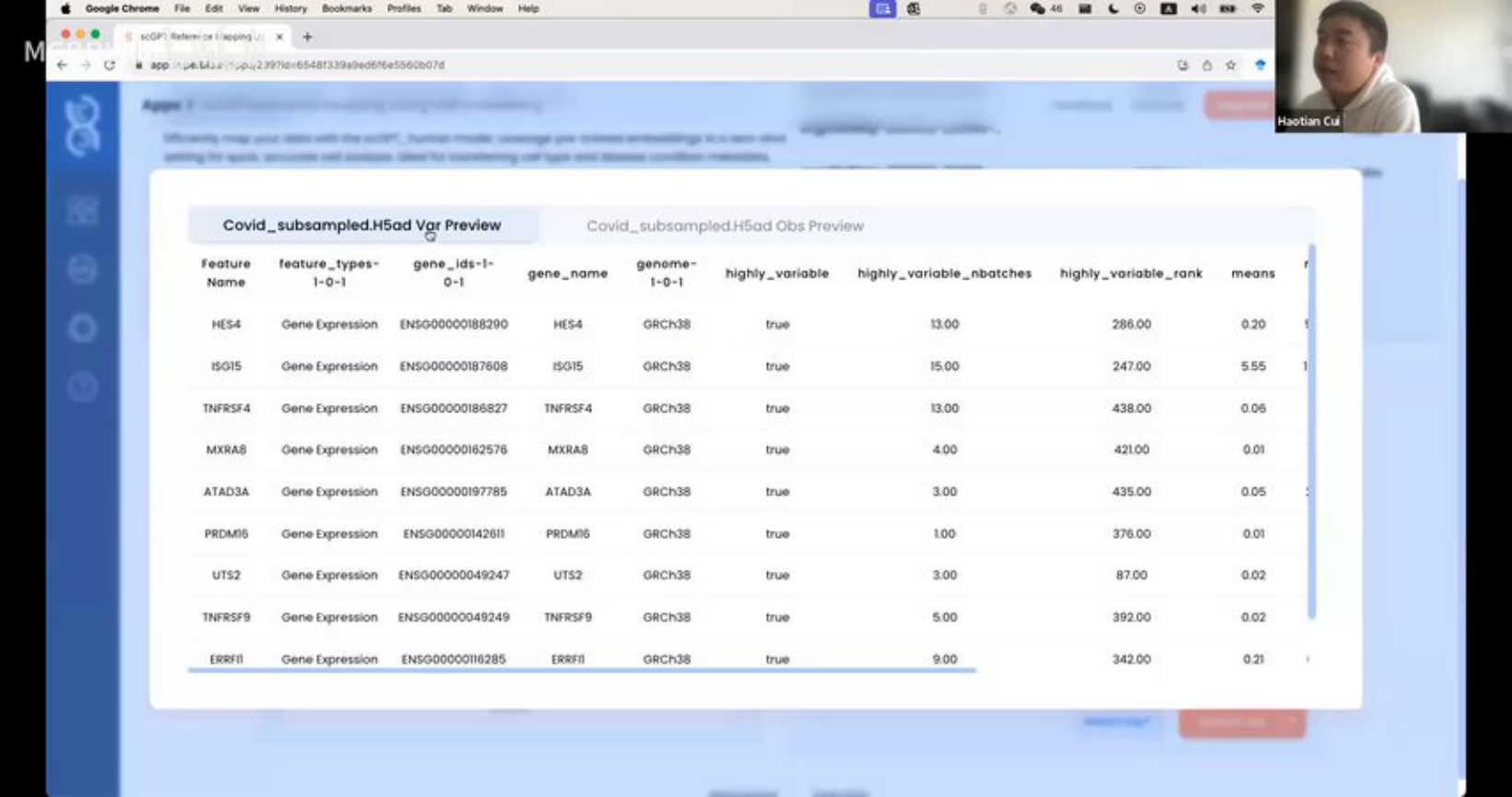
Local Remote

No Files Selected!
Allowed file formats: .h5ad

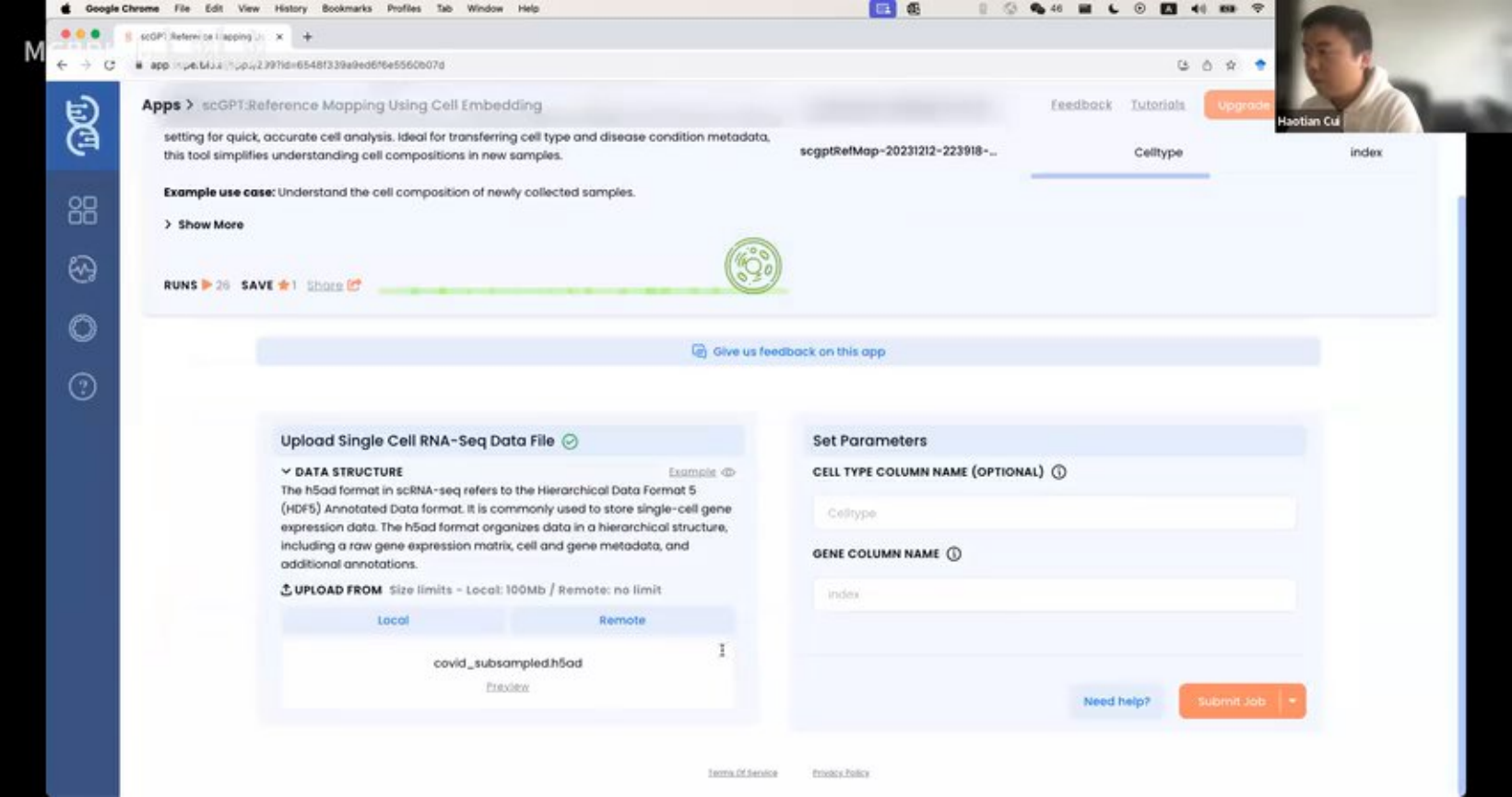
Need format?

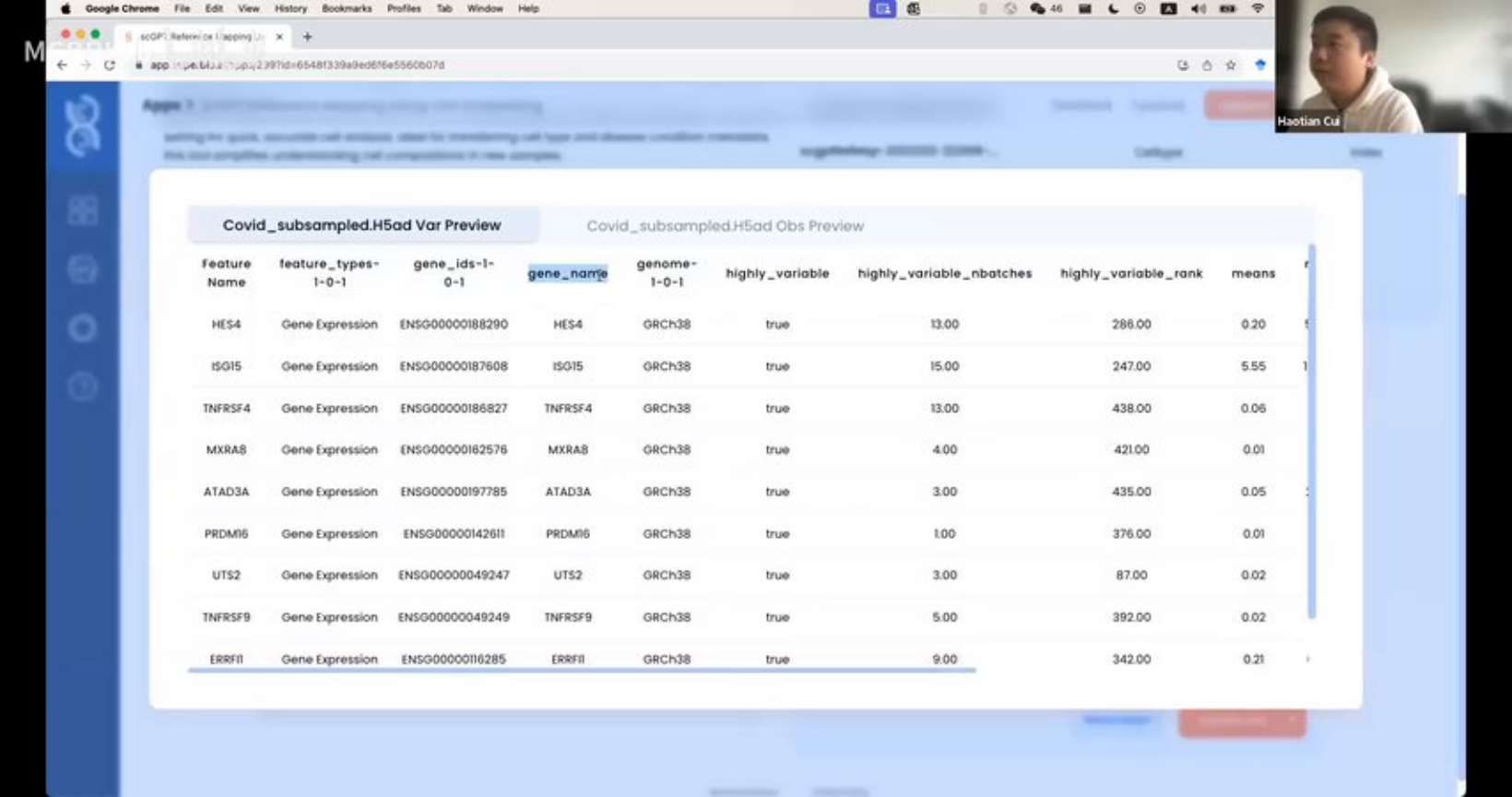
Need help? Submit Job


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Loading Covid_subsampled.H5ad Obs Preview
Loading Covid_subsampled.H5ad Var Preview
Uploading your data, please wait...

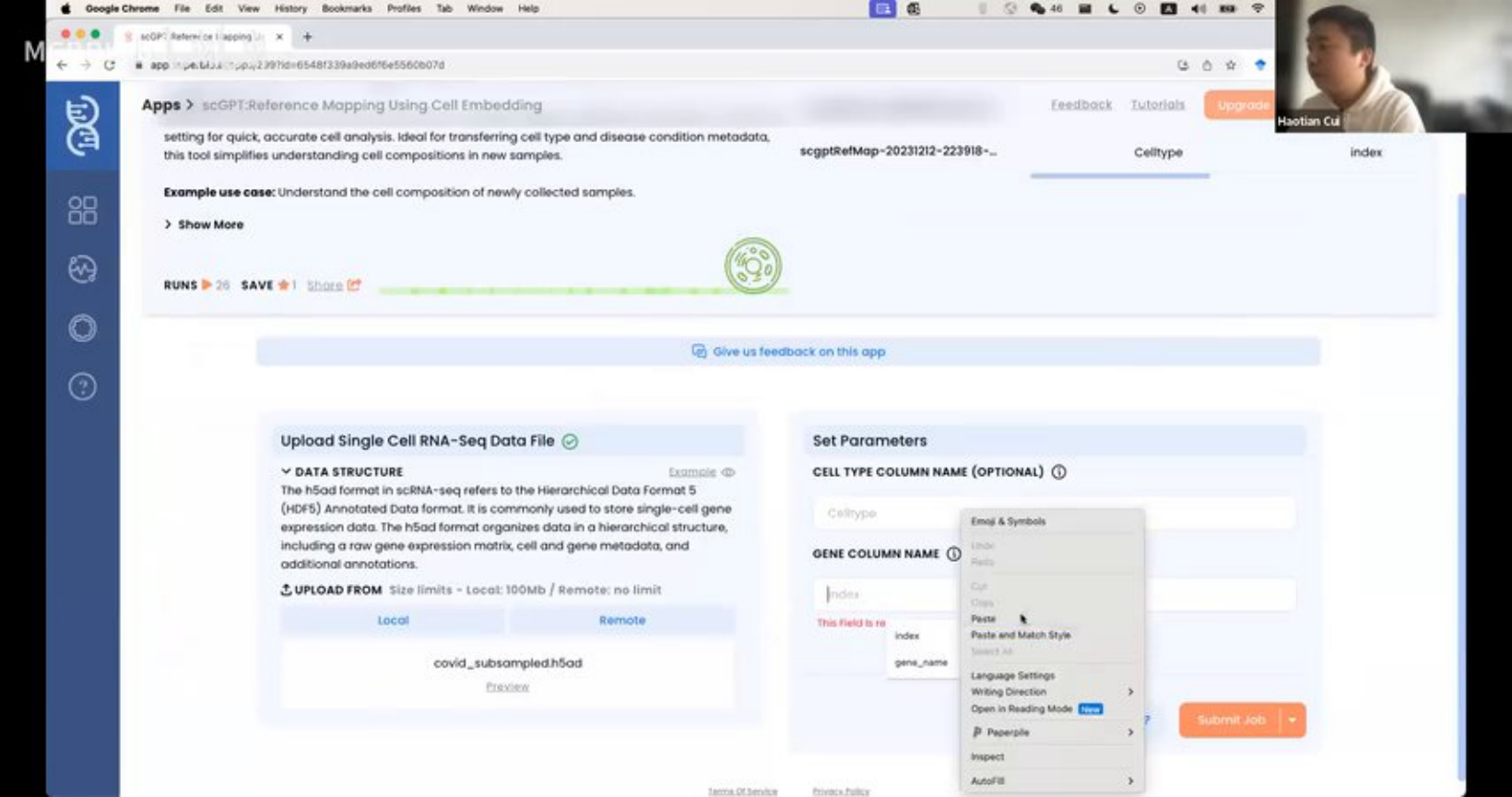


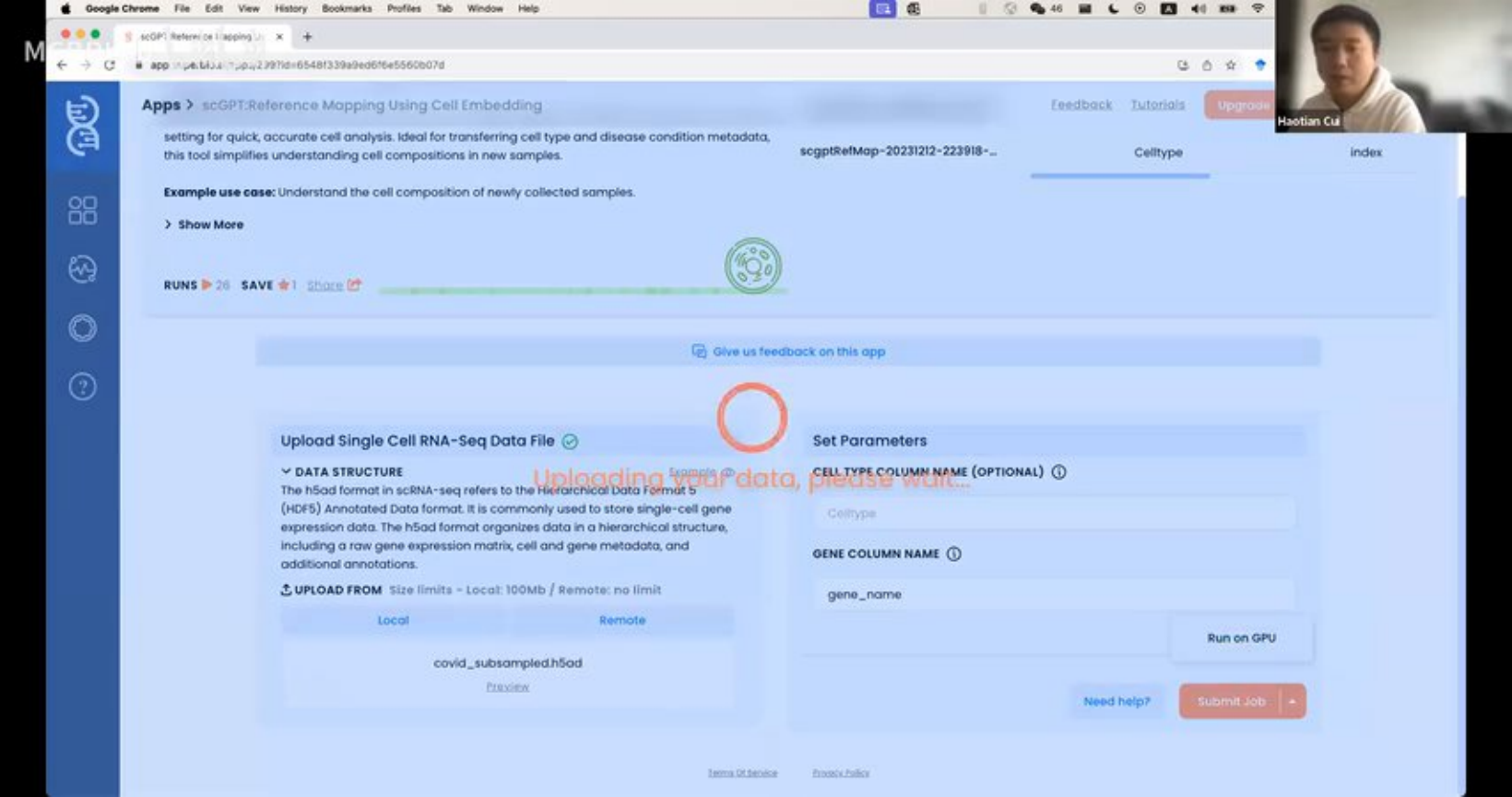
Feature Name	feature_types-1-0-1	gene_ids-1-0-1	gene_name	genome-1-0-1	highly_variable	highly_variable_nbatches	highly_variable_rank	means
HES4	Gene Expression	ENSG00000188290	HES4	GRCh38	true	13.00	286.00	0.20
ISG15	Gene Expression	ENSG00000187608	ISG15	GRCh38	true	15.00	247.00	5.55
TNFRSF4	Gene Expression	ENSG00000186827	TNFRSF4	GRCh38	true	13.00	438.00	0.06
MXRA8	Gene Expression	ENSG00000162576	MXRA8	GRCh38	true	4.00	421.00	0.01
ATAD3A	Gene Expression	ENSG00000197785	ATAD3A	GRCh38	true	3.00	435.00	0.05
PRDM16	Gene Expression	ENSG00000142611	PRDM16	GRCh38	true	1.00	376.00	0.01
UTS2	Gene Expression	ENSG00000049247	UTS2	GRCh38	true	3.00	87.00	0.02
TNFRSF9	Gene Expression	ENSG00000049249	TNFRSF9	GRCh38	true	5.00	392.00	0.02
ERRFI1	Gene Expression	ENSG00000116285	ERRFI1	GRCh38	true	9.00	342.00	0.21

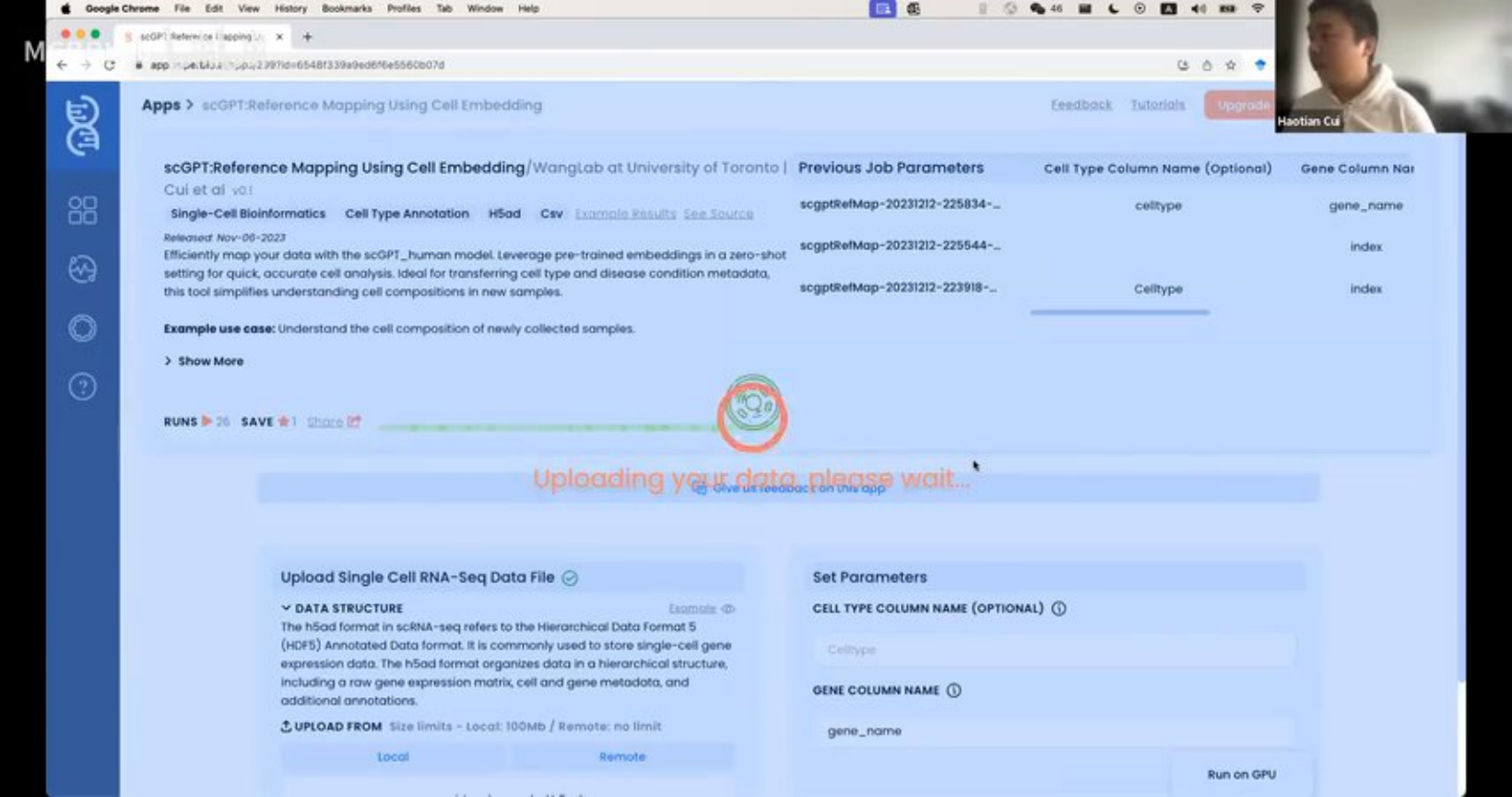

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
scgptRefMap-20231212-223918-...
Celltype
index
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26 SAVE 1 Share
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index
Need help?
Submit Job
Terms Of Service
Privacy Policy

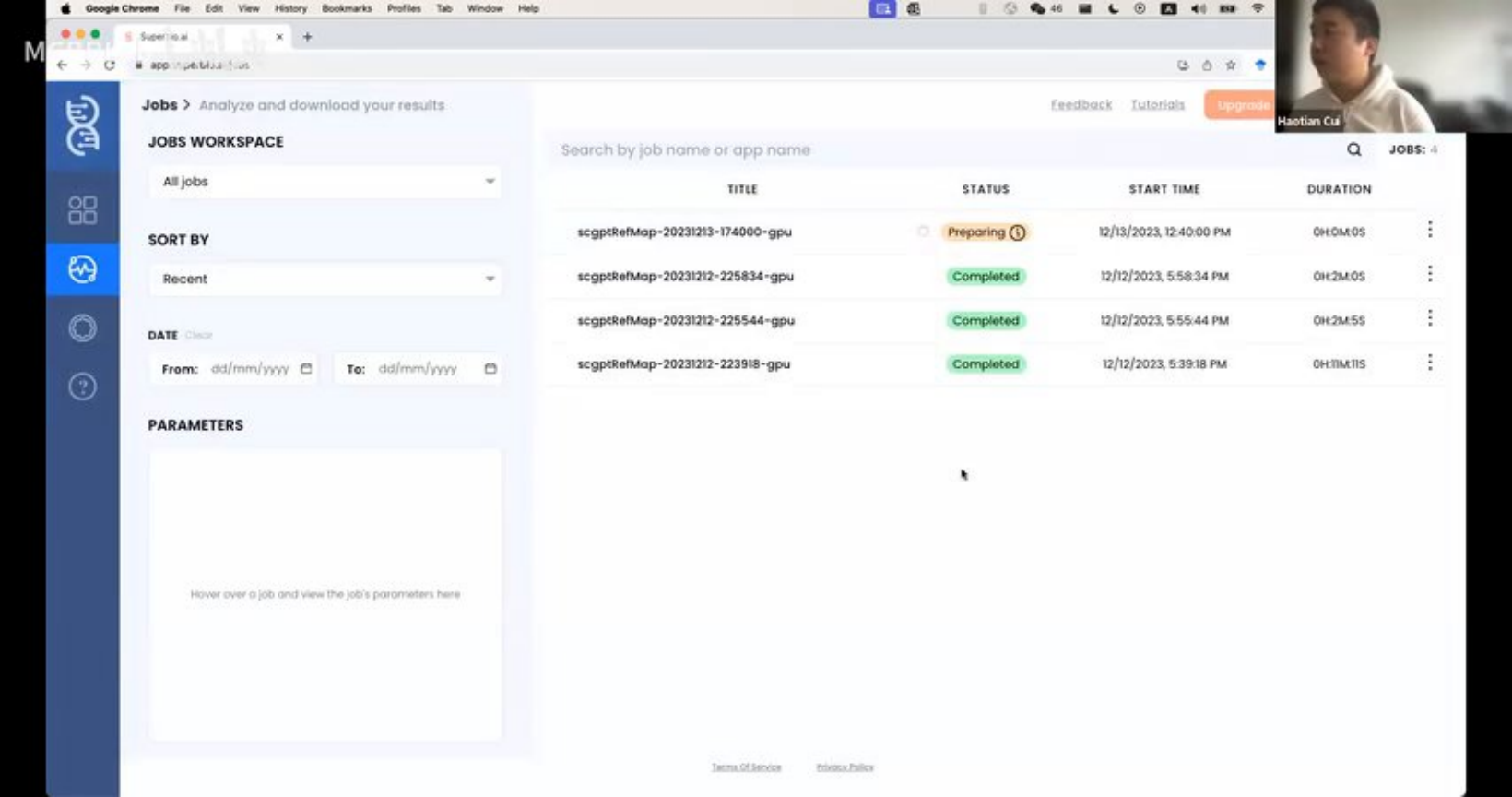


Feature Name	feature_types-1-0-1	gene_ids-1-0-1	gene_name	genome-1-0-1	highly_variable	highly_variable_nbatches	highly_variable_rank	means
HES4	Gene Expression	ENSG00000188290	HES4	GRCh38	true	13.00	286.00	0.20
ISG15	Gene Expression	ENSG00000187608	ISG15	GRCh38	true	15.00	247.00	5.55
TNFRSF4	Gene Expression	ENSG00000186827	TNFRSF4	GRCh38	true	13.00	438.00	0.06
MXRA8	Gene Expression	ENSG00000162576	MXRA8	GRCh38	true	4.00	421.00	0.01
ATAD3A	Gene Expression	ENSG00000197785	ATAD3A	GRCh38	true	3.00	435.00	0.05
PRDM16	Gene Expression	ENSG00000142611	PRDM16	GRCh38	true	1.00	376.00	0.01
UTS2	Gene Expression	ENSG00000049247	UTS2	GRCh38	true	3.00	87.00	0.02
TNFRSF9	Gene Expression	ENSG00000049249	TNFRSF9	GRCh38	true	5.00	392.00	0.02
ERRFI1	Gene Expression	ENSG00000116285	ERRFI1	GRCh38	true	9.00	342.00	0.21


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
scgptRefMap-20231212-223918-...
Celltype
index
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26 SAVE 1 Share
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
index
This field is re
index
gene_name
Emoji & Symbols
Undo
Redo
Cut
Copy
Paste
Paste and Match Style
Select All
Language Settings
Writing Direction
Open in Reading Mode
New
Paperpile
Inspect
AutoFill
Submit Job
Terms Of Service
Privacy Policy


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback
Tutorials
Upgrade
setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
scgptRefMap-20231212-223918-...
Celltype
index
Example use case: Understand the cell composition of newly collected samples.
Show More
RUNS 26 SAVE 1 Share
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
covid_subsampled.h5ad
Preview
Uploading your data, please wait...
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
gene_name
Run on GPU
Need help?
Submit Job
Terms Of Service
Privacy Policy


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding
Feedback Tutorials Upgrade
scGPT:Reference Mapping Using Cell Embedding/WangLab at University of Toronto | Cui et al v0.1
Single-Cell Bioinformatics Cell Type Annotation H5ad Csv Example Results See Source
Released: Nov-06-2023
Efficiently map your data with the scGPT_human model. Leverage pre-trained embeddings in a zero-shot setting for quick, accurate cell analysis. Ideal for transferring cell type and disease condition metadata, this tool simplifies understanding cell compositions in new samples.
Example use case: Understand the cell composition of newly collected samples.
> Show More
Previous Job Parameters
Cell Type Column Name (Optional)
Gene Column Nar
scgptRefMap-20231212-225834-...
celltype
gene_name
scgptRefMap-20231212-225544-...
index
scgptRefMap-20231212-223918-...
Celltype
index
RUNS 26 SAVE 1 Share
Uploading your data, please wait...
Give us feedback on this app
Upload Single Cell RNA-Seq Data File
DATA STRUCTURE
Example
The h5ad format in scRNA-seq refers to the Hierarchical Data Format 5 (HDF5) Annotated Data format. It is commonly used to store single-cell gene expression data. The h5ad format organizes data in a hierarchical structure, including a raw gene expression matrix, cell and gene metadata, and additional annotations.
UPLOAD FROM Size limits - Local: 100Mb / Remote: no limit
Local
Remote
Set Parameters
CELL TYPE COLUMN NAME (OPTIONAL)
Celltype
GENE COLUMN NAME
gene_name
Run on GPU


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Jobs > Analyze and download your results
Feedback
Tutorials
Upgrade
JOBS WORKSPACE
All jobs
SORT BY
Recent
DATE
From: dd/mm/yyyy
To: dd/mm/yyyy
PARAMETERS
Hover over a job and view the job's parameters here
Search by job name or app name
JOBS: 4
TITLE
STATUS
START TIME
DURATION
scgptRefMap-20231213-174000-gpu
Preparing
12/13/2023, 12:40:00 PM
0H:0M:0S
scgptRefMap-20231212-225834-gpu
Completed
12/12/2023, 5:58:34 PM
0H:2M:0S
scgptRefMap-20231212-225544-gpu
Completed
12/12/2023, 5:55:44 PM
0H:2M:5S
scgptRefMap-20231212-223918-gpu
Completed
12/12/2023, 5:39:18 PM
0H:11M:11S
Terms Of Service
Privacy Policy

Results

Below, you'll find a preview of the comparison table showcasing the original labels alongside the predicted labels if the original cell type column is provided in the parameters before submitting the job. For a comprehensive view, you can download the complete

Parameters

File H5ad:	Covid_subsampled.H5ad
Cell Type Column Name (Optional):	Celltype
Gene Column Name:	Gene_name
Workflow_name:	Workflow:Yml

Actions

Download Output Files

Share This Job

Reopen App

Save App To My Library

Give us feedback on this app

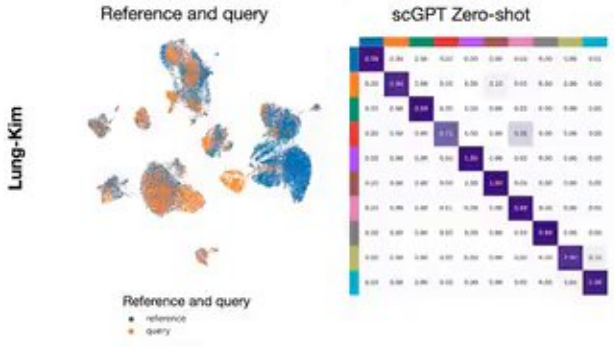
Propagated CellXGene labels (Preview)

Original Labels	CellXGene Labels
Monocytes	classical monocyte
Mast cells	mast cell
AT2	type II pneumocyte
T cell	effector memory CD8-positive, alpha-beta T cell
CD4+ T cells	CD4-positive, alpha-beta memory T cell
AT2	type II pneumocyte
Macrophages	macrophage

Contents

Zero-shot reference mapping

- We also use the embeddings in a reference mapping manner
- We split the data to simulate a setting that some of the patient cells, query set, were not annotated. Cell types were propagated from reference data set to label the query set.
- The zero-shot workflow demonstrates competitive accuracy across comparisons.



Dataset	Method	Evaluation Metrics			
		Accuracy	*Precision*	*Recall*	*MacroF1*
COVID-19	scGPT (fine-tuned)	**0.891**	**0.577**	**0.567**	**0.547**
	scGPT (zero-shot)	0.867	0.513	0.476	0.468
	Azimuth	0.878	0.515	0.418	0.444
	expiMap	0.730	0.441	0.325	0.335
Lung-Kim	scGPT (fine-tuned)	**0.974**	0.959	**0.967**	**0.962**
	scGPT (zero-shot)	0.968	**0.970**	0.933	0.948
	Azimuth	0.970	0.957	0.962	0.959
	expiMap	0.920	0.940	0.846	0.878

Going forward

- Zero-shot can be promising, we are going to provide further updates for the scenario:
  - Hierarchical cell type labeling of reference mapping from atlas
  - Going to provide new checkpoint specifically for zero-shot cell embeddings
  - Zero-shot applications using gene embeddings
  - Others

MCBRLab

Acknowledgement

Thank you!

scGPT codebase: https://github.com/bowang-lab/scGPT.

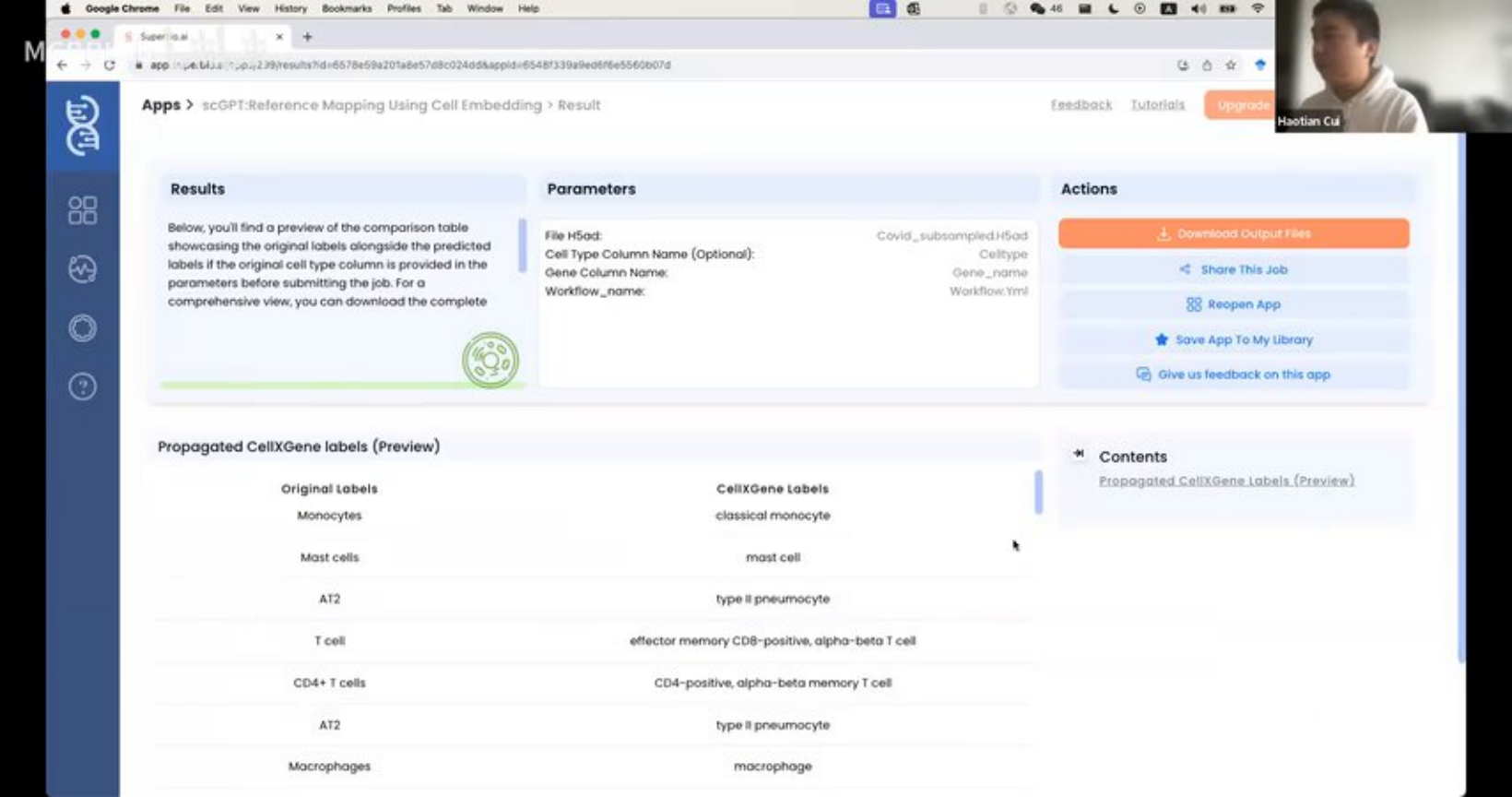
Sincere gratitude to collaborators: Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Dr. Lin Zhang, Dr. Nan Duan, and Dr. Bo Wang

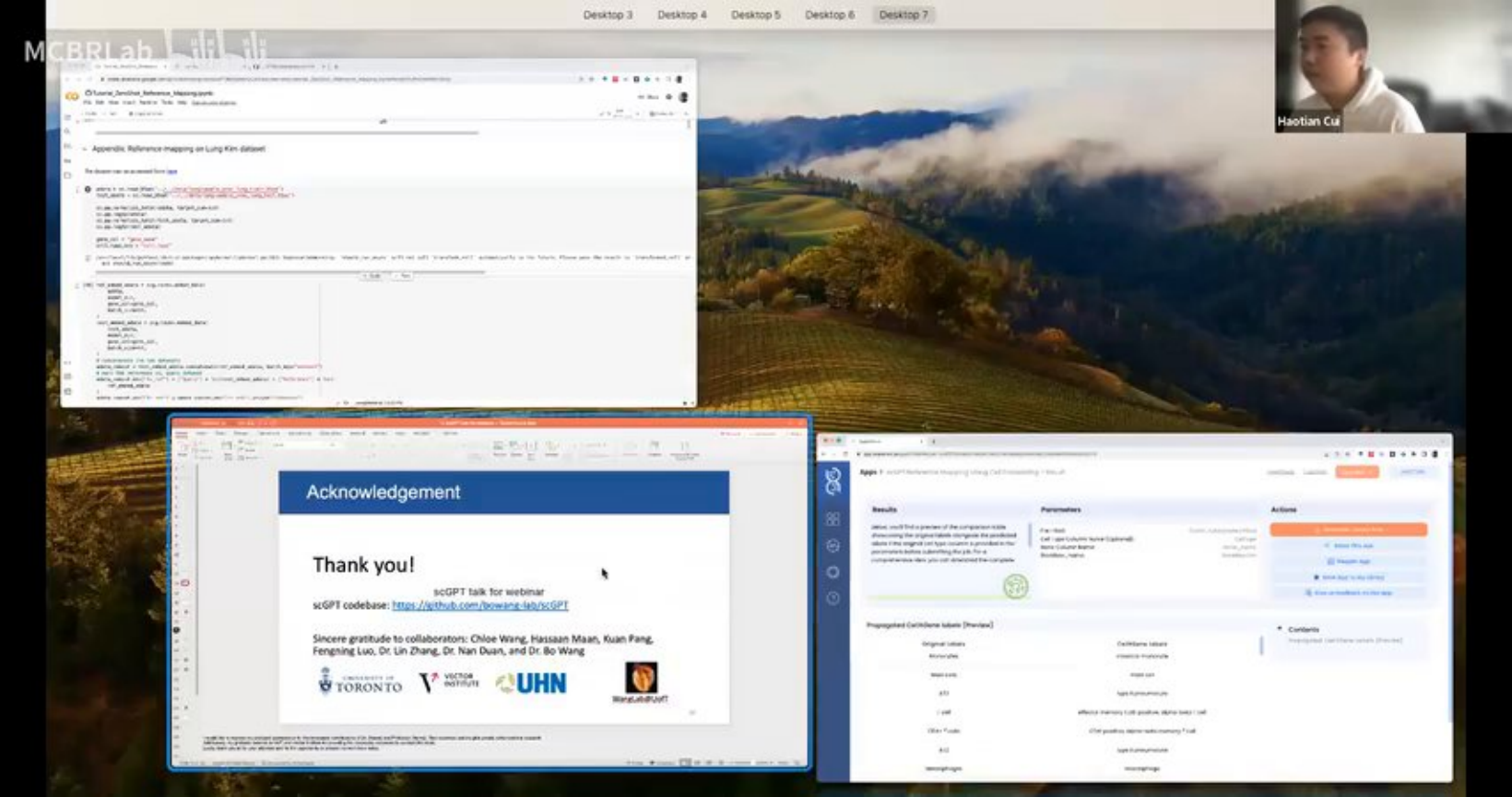
UNIVERSITY OF TORONTO

VECTOR INSTITUTE

UHN

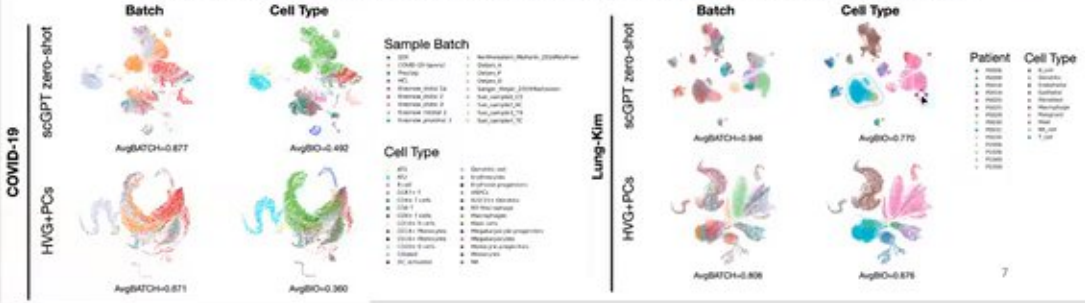
WangLab@UofT


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Haotian Cui
Apps > scGPT:Reference Mapping Using Cell Embedding > Result
Feedback
Tutorials
Upgrade
Results
Below, you'll find a preview of the comparison table showcasing the original labels alongside the predicted labels if the original cell type column is provided in the parameters before submitting the job. For a comprehensive view, you can download the complete
Parameters
File H5ad: Covid_subsampled.H5ad
Cell Type Column Name (Optional): Celltype
Gene Column Name: Gene_name
Workflow_name: Workflow.Yml
Actions
Download Output Files
Share This Job
Reopen App
Save App To My Library
Give us feedback on this app
Propagated CellXGene labels (Preview)
Original Labels
CellXGene Labels
Monocytes
classical monocyte
Mast cells
mast cell
AT2
type II pneumocyte
T cell
effector memory CD8-positive, alpha-beta T cell
CD4+ T cells
CD4-positive, alpha-beta memory T cell
AT2
type II pneumocyte
Macrophages
macrophage
Contents
Propagated CellXGene Labels (Preview)


Desktop 3
Desktop 4
Desktop 5
Desktop 6
Desktop 7
MCBRLab
Haotian Cui
Appendix: Reference mapping on Lung Kim dataset
Acknowledgement
Thank you!
scGPT talk for webinar
scGPT codebase: https://github.com/bowang-lab/scGPT
Sincere gratitude to collaborators: Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Dr. Lin Zhang, Dr. Nan Duan, and Dr. Bo Wang
UNIVERSITY OF TORONTO
VECTOR INSTITUTE
UHN
Results
Parameters
Actions
Contents

Zero-shot applications

- **Zero-shot**, meaning using the pretrained model to generate embeddings for new data, without any further training, **much faster, more accessible**
- UMAP of zero-shot embeddings on two disease datasets, compared to HVGs
  - Considerable ability to distinguish cell types
  - Note the pretraining process is not designed for mitigating technical batch effects



Have been working new extensions exploring the possibility of zero-shot applications. Here, I'd like to share some new results

Particularly want to mention the influence of pretraining,
so of the technical info is actually signals


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai


MCBRLab
bilibili
superbio
Berke - Superbio.ai