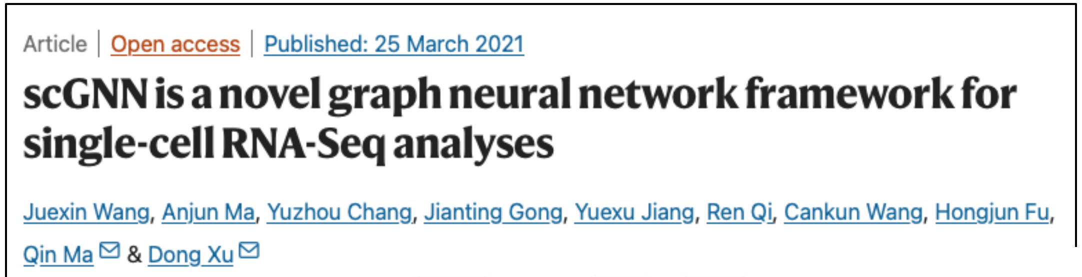
Single-cell RNA-seq dataset acquisition, model training, and analysis using scGNN+ integrated ChatGPT

Yi Jiang
Graduate Research Assistant
Department of Biomedical Informatics
The Ohio State University
12/2/2024

THE OHIO STATE UNIVERSITY
WEXNER MEDICAL CENTER

BMBL
Bioinformatics and Mathematical Biosciences Lab

scGNN series work



Article | Open access | Published: 25 March 2021

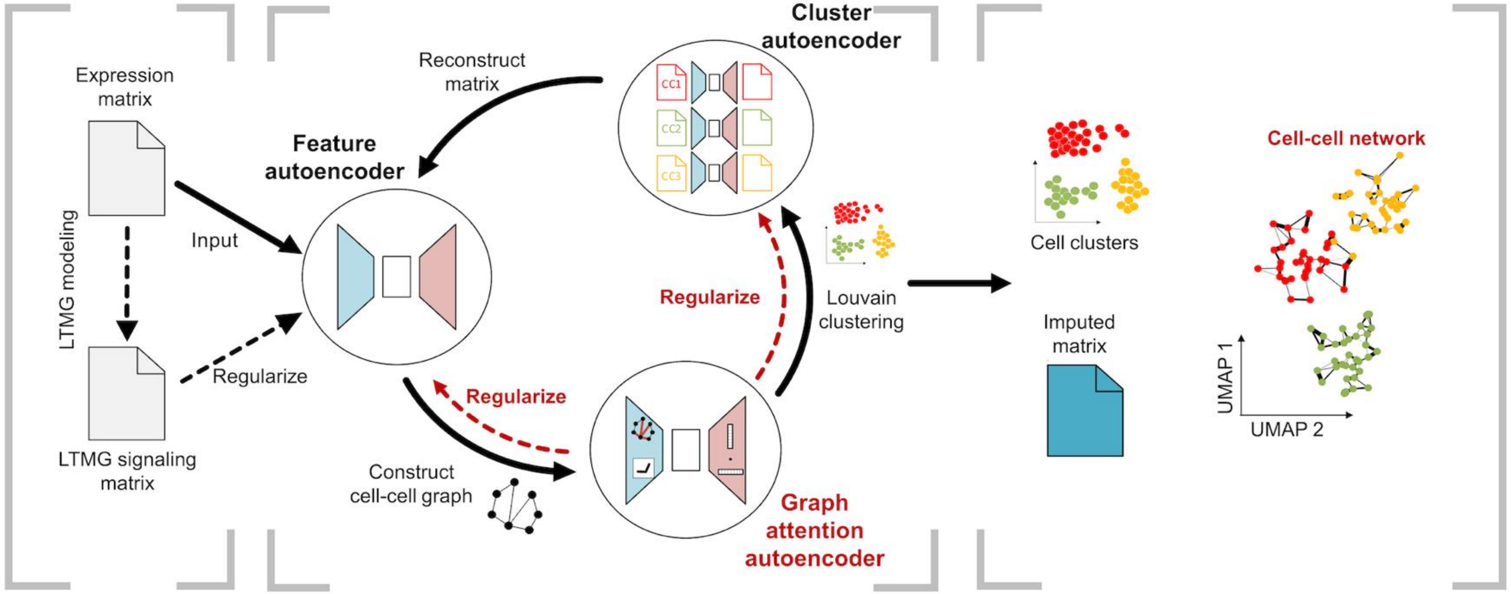
scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses

Juexin Wang, Anjun Ma, Yuzhou Chang, Jianting Gong, Yuexu Jiang, Ren Qi, Cankun Wang, Hongjun Fu, Qin Ma & Dong Xu

Nature Communications

Cited by 396

In-house graph neural network for cellular heterogeneity

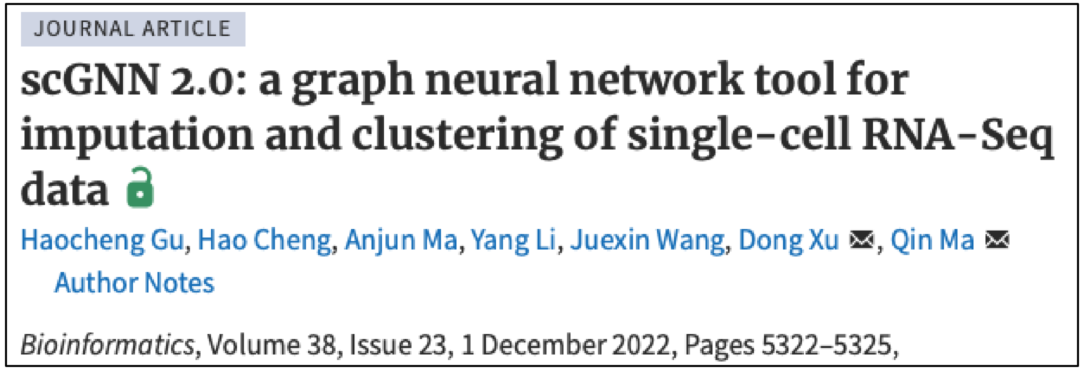


- The first graph neural network for scRNA-seq data;
- Consider neighbor effects in characterizing cell-cell relations;

- Simultaneously perform cell clustering and gene imputation;
- Achieved better performance than existing tools;
- scGNN has been cited 260 times in two years.

Juexin Wang*, Anjun Ma*, et al., scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nature Communications*. (2021)

scGNN series work



JOURNAL ARTICLE

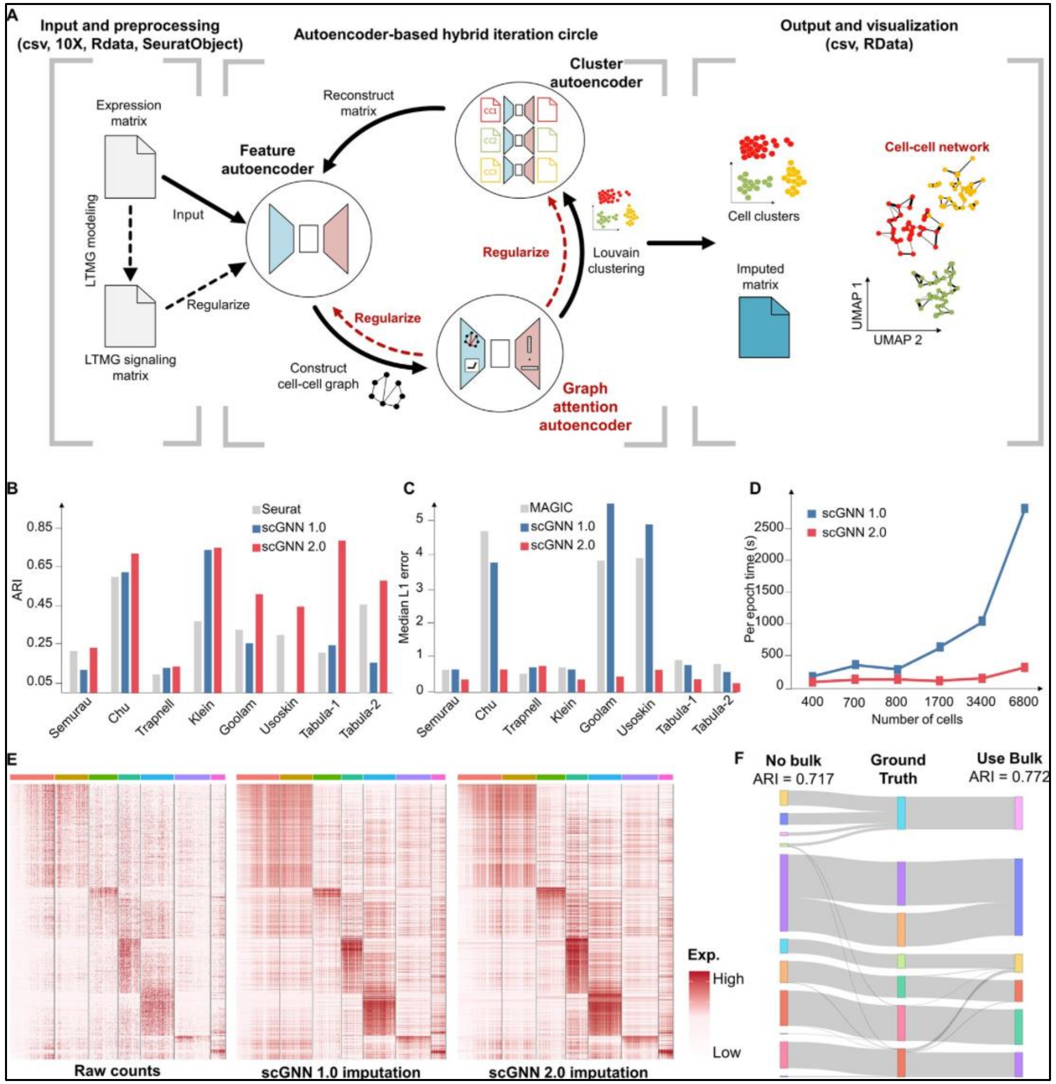
scGNN 2.0: a graph neural network tool for imputation and clustering of single-cell RNA-Seq data

Haocheng Gu, Hao Cheng, Anjun Ma, Yang Li, Juexin Wang, Dong Xu ✉, Qin Ma ✉

Author Notes

Bioinformatics, Volume 38, Issue 23, 1 December 2022, Pages 5322–5325,

- Compared to the previous version, the 2.0 framework adds
  - New regularizations
  - The use of graph multi-head attention mechanism
  - UMAP-based cell-cell network visualization

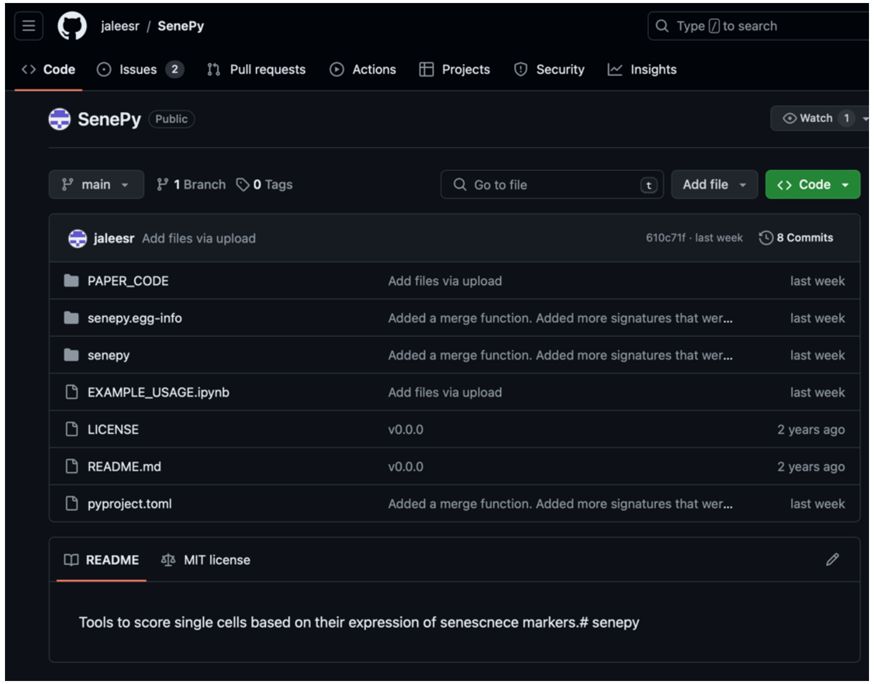


Outline

- ✓ Motivation of scGNN+
- ChatGPT-combined Framework
- scGNN+ applications in Single-cell RNA-seq dataset

The motivation of scGNN 3.0

- There are many issues with making applications of many biomedical algorithm packages, tools, or scripts, even if they have tutorials.
  - Many tutorials are designed with too many programming techniques or casual introductions for their pipelines. It will be hard to help biologists use these algorithms to analyze their data.
  - The qualities of tutorials are totally based on the consciousness of developer and supervision of paper reviewers.
- For example:

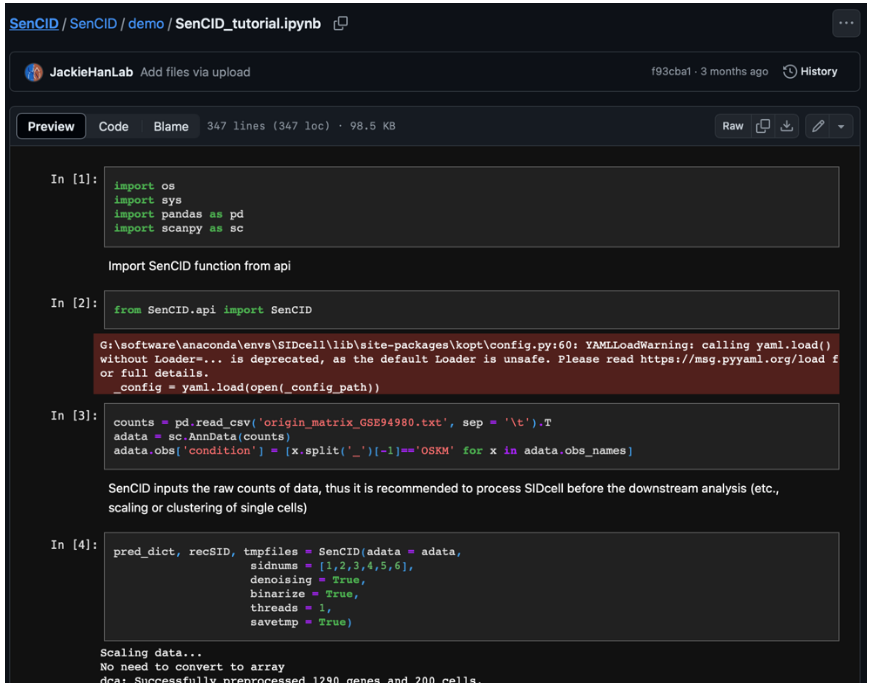


jaleesr Add files via upload 610c71f · last week 8 Commits

PAPER_CODE	Add files via upload	last week
senepy.egg-info	Added a merge function. Added more signatures that wer...	last week
senepy	Added a merge function. Added more signatures that wer...	last week
EXAMPLE_USAGE.ipynb	Add files via upload	last week
LICENSE	v0.0.0	2 years ago
README.md	v0.0.0	2 years ago
pyproject.toml	Added a merge function. Added more signatures that wer...	last week

Tools to score single cells based on their expression of senescnece markers.# senepy

Sanborn, Mark A., et al. bioRxiv (2023)


SenCID / SenCID / demo / SenCID_tutorial.ipynb
JackieHanLab Add files via upload
f93cba1 · 3 months ago
History
Preview
Code
Blame
347 lines (347 loc) · 98.5 KB
Raw

In [1]:

```
import os
import sys
import pandas as pd
import scanpy as sc
```

Import SenCID function from api

In [2]:

```
from SenCID.api import SenCID
```

```
G:\software\anaconda\envs\SIDcell\lib\site-packages\kopt\config.py:60: YAMLLoadWarning: calling yaml.load()
without Loader=... is deprecated, as the default Loader is unsafe. Please read https://msg.pyyaml.org/load f
or full details.
  _config = yaml.load(open(_config_path))
```

In [3]:

```
counts = pd.read_csv('origin_matrix_GSE94980.txt', sep = '\t').T
adata = sc.AnnData(counts)
adata.obs['condition'] = [x.split('_')[-1]=='OSKM' for x in adata.obs_names]
```

SenCID inputs the raw counts of data, thus it is recommended to process SIDcell before the downstream analysis (etc., scaling or clustering of single cells)

In [4]:

```
pred_dict, recSID, tmpfiles = SenCID(adata = adata,
                    sidnums = [1,2,3,4,5,6],
                    denoising = True,
                    binarize = True,
                    threads = 1,
                    savetmp = True)
```

```
Scaling data...
No need to convert to array
dca: Successfully preprocessed 1290 genes and 200 cells.
```

Tao, Wanyu, et. al. Cell Metabolism (2024)

The motivation of scGNN 3.0

- For the easiest way to avoid this, researchers usually build a web server for the specific tools where the user just provides their input, and then they can get the output.
  - There are many issues with this solution, which include
    - Server maintenance (almost 80% of developed servers cannot work because of absenting maintenance)
    - Analysis missing (only provide predicted results but absenting analysis mentioned in their paper)
    - Weak to answer users' questions according to output results.
    - The volume of work is huge with non-efficiency.
    - …

404

Sorry, the page you are looking for does not exist.

The motivation of scGNN 3.0

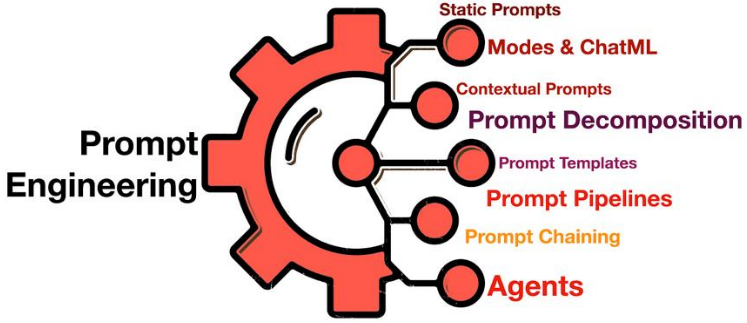
Large Language Models (LLMs), such as ChatGPT, have the capability to understand users' questions about

- Tutorials (understand the logical introductions of pipelines)
- Arti
- Co
- …

By using ChatGPT and prompting engineering

- **Establish a tool to assist in users' understanding of scGNN series tools, generate some analysis codes, and help run them.**
- **Apply this tool to any project to test whether their tutorials can be understood by ChatGPT with prompt engineering**
- **Provide a way to assist in any project to make them user-friendly.**

Some … nts under the provided materials, like tutorials.



https://cobusgreyling.medium.com/eight-prompt-engineering-implementations-fc361fdc87b

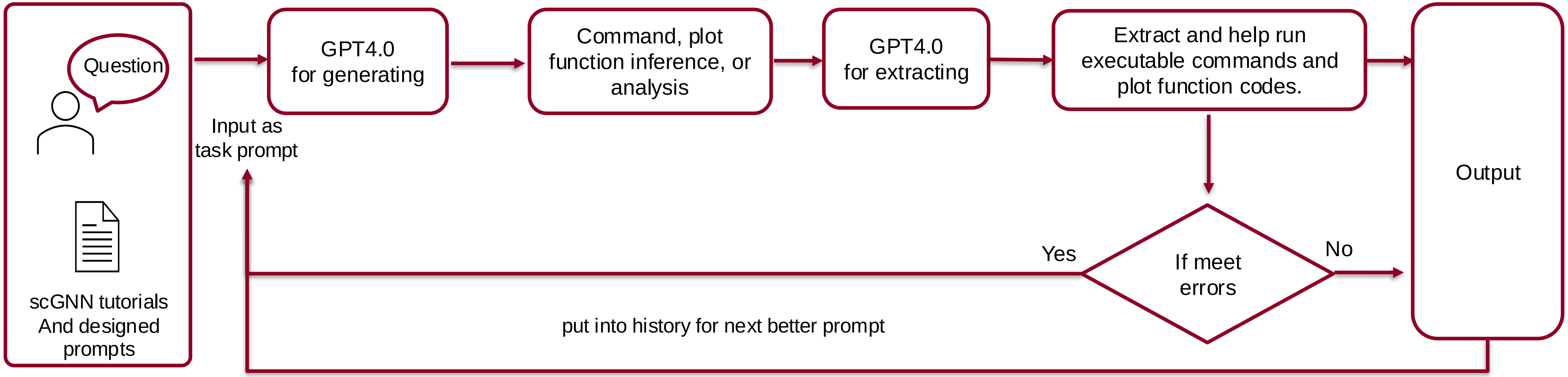
Outline

- Motivation of scGNN-3.0

✓ ChatGPT-combined Framework

- scGNN+ applications in Single-cell RNA-seq dataset

scGNN3.0 is coming!



- Use two GPTs, one is to generate commands, plot function codes, and provide analysis, the other one is to extract executable commands or plot function codes with a comprehensive prompt.
- The prompt list of every output is saved as chat history to let GPTs generate codes or commands based on chat context.

GPTs + Tutorial prompt + Auto-executing = Chat with scGNN!
https://bmblx.bmi.osumc.edu/scgnn3

The standard for a function code statement

1. Function Name: The unique identifier used to call the function.
2. Description: A brief explanation of what the function does or is used for.

Function detail = A (Overview introduction) + B (Input parameters introduction) + C (Return/results) + D (Use examples) + E (Additional notes)

A (Overview introduction)	B (Input parameter introduction)	C: Return/results	D: Use examples	E: Additional notes
Function name and brief description	Name, data type and structure (if some data is input), description, default value, optional status, and whether have limitation or cooperation with other parameters	returned file name, address, returned data format and structure		Constraints, version and errors

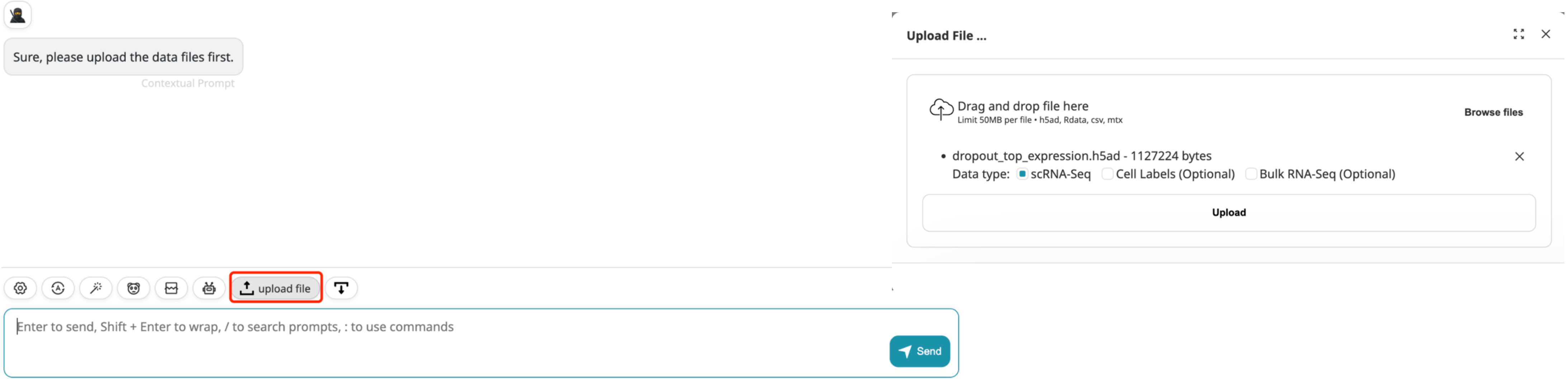
- Version: The version of the software or library the function belongs to, if relevant.

Outline

- Motivation of scGNN-3.0
- ChatGPT-combined Framework
- ✓ scGNN+ applications in Single-cell RNA-seq dataset

An example by using scGNN3.0-server

1. upload your data



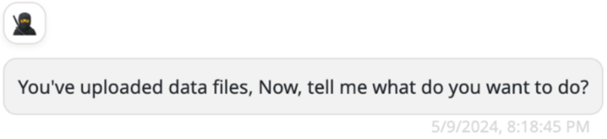
Sure, please upload the data files first.

- dropout_top_expression.h5ad - 1127224 bytes

Data type: ● scRNA-Seq ☐ Cell Labels (Optional) ☐ Bulk RNA-Seq (Optional)

Enter to send, Shift + Enter to wrap, / to search prompts, : to use commands

2. When your dataset is ready, the chat robot will guide you to provide some order

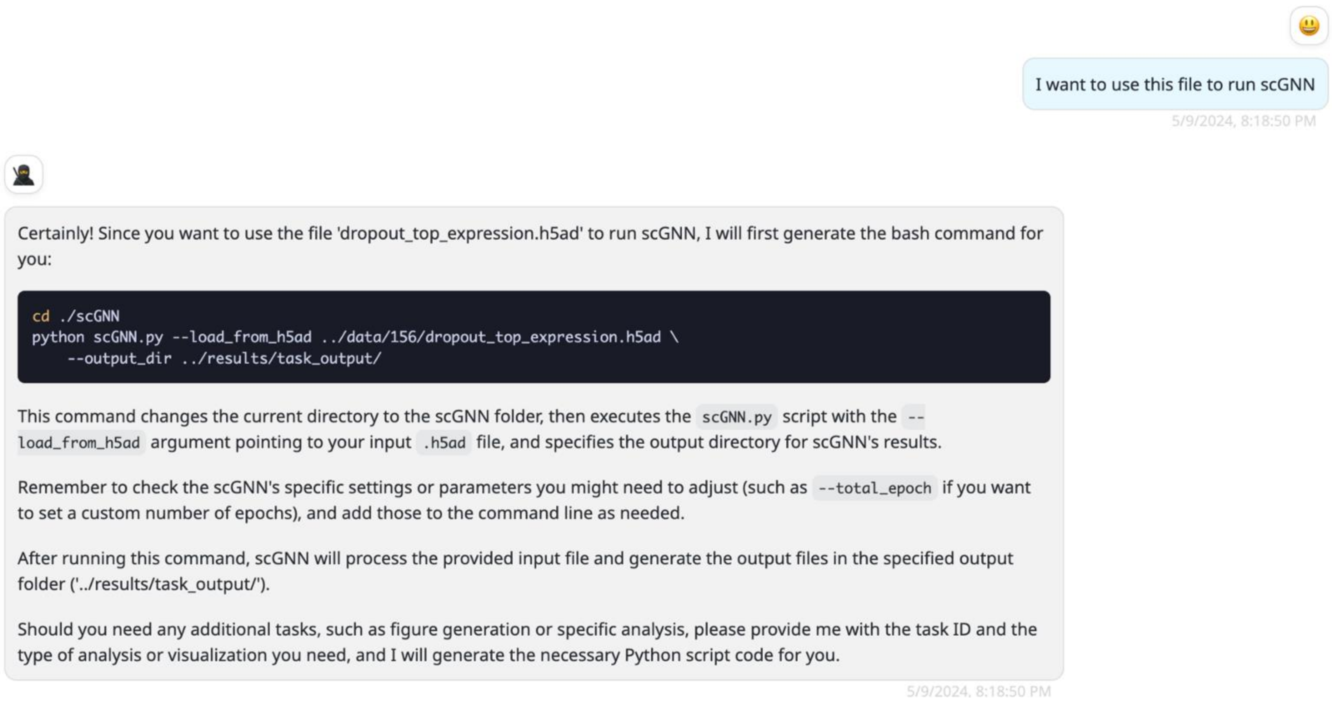


You've uploaded data files, Now, tell me what do you want to do?

5/9/2024, 8:18:45 PM

An example by using scGNN3.0-server

3. you can run scGNN by using this uploaded data



I want to use this file to run scGNN

5/9/2024, 8:18:50 PM

Certainly! Since you want to use the file 'dropout_top_expression.h5ad' to run scGNN, I will first generate the bash command for you:

```
cd ./scGNN
python scGNN.py --load_from_h5ad ../data/156/dropout_top_expression.h5ad \
    --output_dir ../results/task_output/
```

This command changes the current directory to the scGNN folder, then executes the `scGNN.py` script with the `--load_from_h5ad` argument pointing to your input `.h5ad` file, and specifies the output directory for scGNN's results.

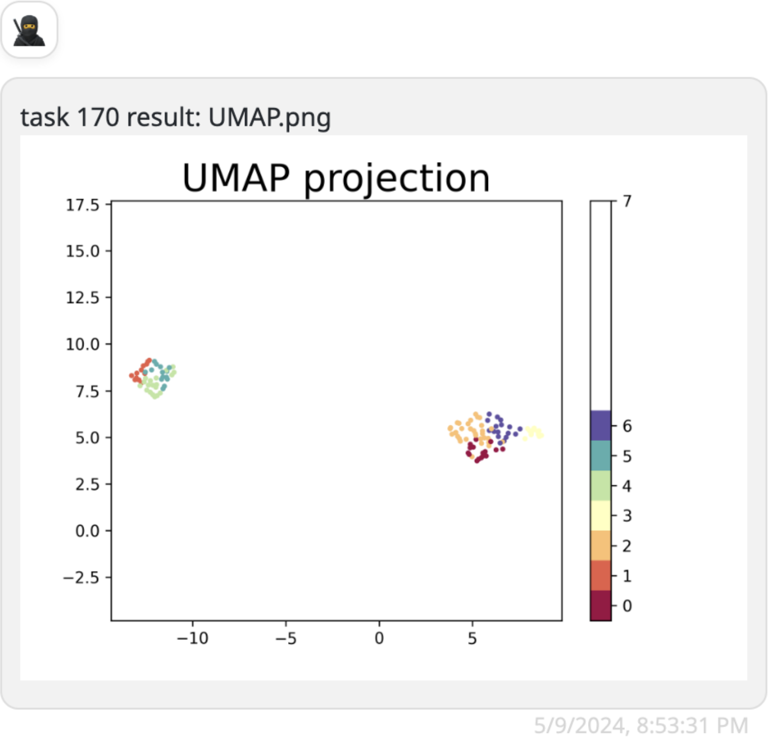
Remember to check the scGNN's specific settings or parameters you might need to adjust (such as `--total_epoch` if you want to set a custom number of epochs), and add those to the command line as needed.

After running this command, scGNN will process the provided input file and generate the output files in the specified output folder ('../results/task_output/').

Should you need any additional tasks, such as figure generation or specific analysis, please provide me with the task ID and the type of analysis or visualization you need, and I will generate the necessary Python script code for you.

5/9/2024, 8:18:50 PM

4. When the task results are ready, the UMAP figure will be provided automatically

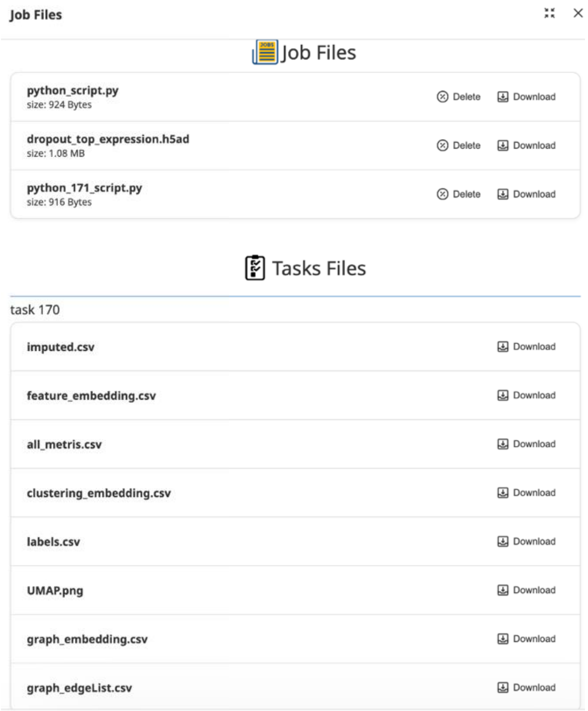


task 170 result: UMAP.png

5/9/2024, 8:53:31 PM

An example by using scGNN3.0-server

5. Now, you can download the result data of scGNN output



Job Files

Job Files

python_script.py
size: 924 Bytes
Delete Download

dropout_top_expression.h5ad
size: 1.08 MB
Delete Download

python_171_script.py
size: 916 Bytes
Delete Download

Tasks Files

task 170

imputed.csv Download

feature_embedding.csv Download

all_metris.csv Download

clustering_embedding.csv Download

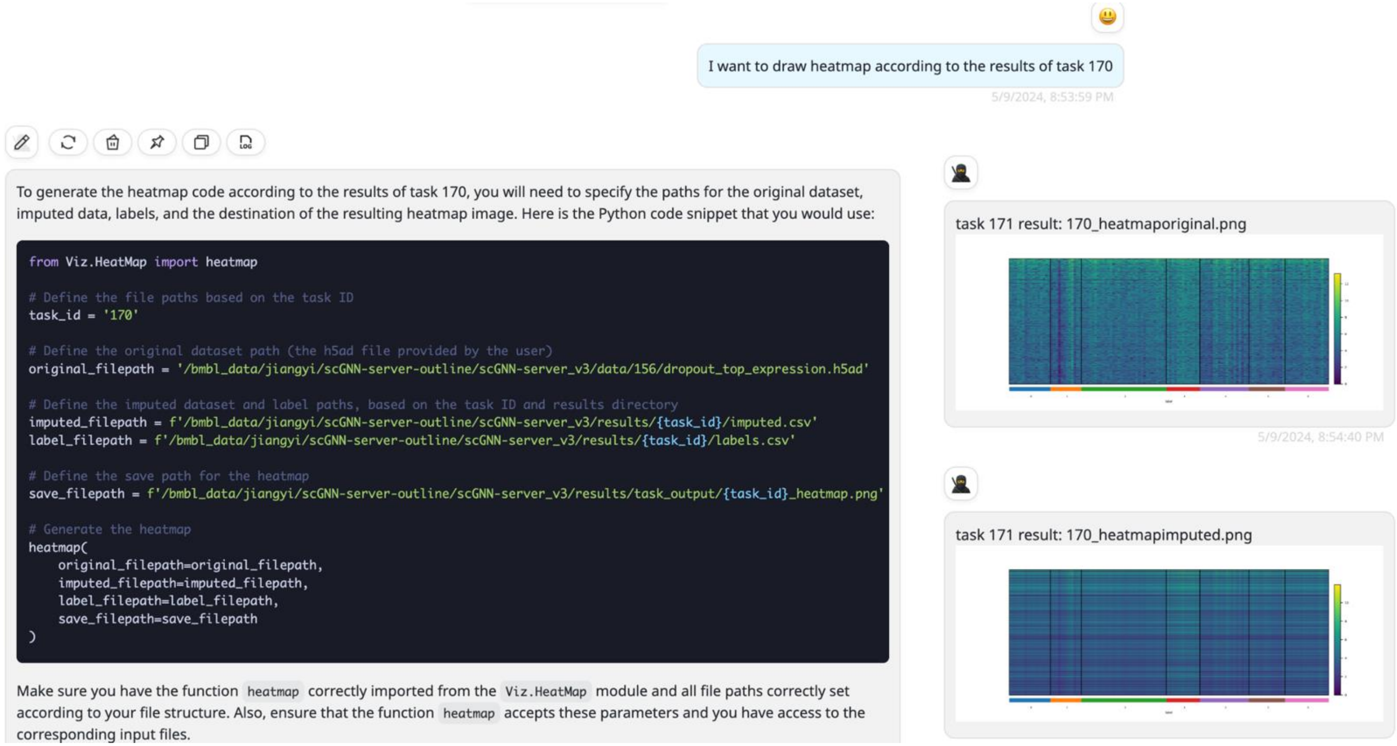
labels.csv Download

UMAP.png Download

graph_embedding.csv Download

graph_edgeList.csv Download

6. The results are ready, you probably want to see some analysis figure: Ask the chat robot to draw a Heatmap figure by using the output of scGNN



I want to draw heatmap according to the results of task 170

5/9/2024, 8:53:59 PM

To generate the heatmap code according to the results of task 170, you will need to specify the paths for the original dataset, imputed data, labels, and the destination of the resulting heatmap image. Here is the Python code snippet that you would use:

```
from Viz.HeatMap import heatmap

# Define the file paths based on the task ID
task_id = '170'

# Define the original dataset path (the h5ad file provided by the user)
original_filepath = '/bmbl_data/jiangyi/scGNN-server-outline/scGNN-server_v3/data/156/dropout_top_expression.h5ad'

# Define the imputed dataset and label paths, based on the task ID and results directory
imputed_filepath = f'/bmbl_data/jiangyi/scGNN-server-outline/scGNN-server_v3/results/{task_id}/imputed.csv'
label_filepath = f'/bmbl_data/jiangyi/scGNN-server-outline/scGNN-server_v3/results/{task_id}/labels.csv'

# Define the save path for the heatmap
save_filepath = f'/bmbl_data/jiangyi/scGNN-server-outline/scGNN-server_v3/results/task_output/{task_id}_heatmap.png'

# Generate the heatmap
heatmap(
    original_filepath=original_filepath,
    imputed_filepath=imputed_filepath,
    label_filepath=label_filepath,
    save_filepath=save_filepath
)
```

Make sure you have the function `heatmap` correctly imported from the `Viz.HeatMap` module and all file paths correctly set according to your file structure. Also, ensure that the function `heatmap` accepts these parameters and you have access to the corresponding input files.

task 171 result: 170_heatmaporiginal.png

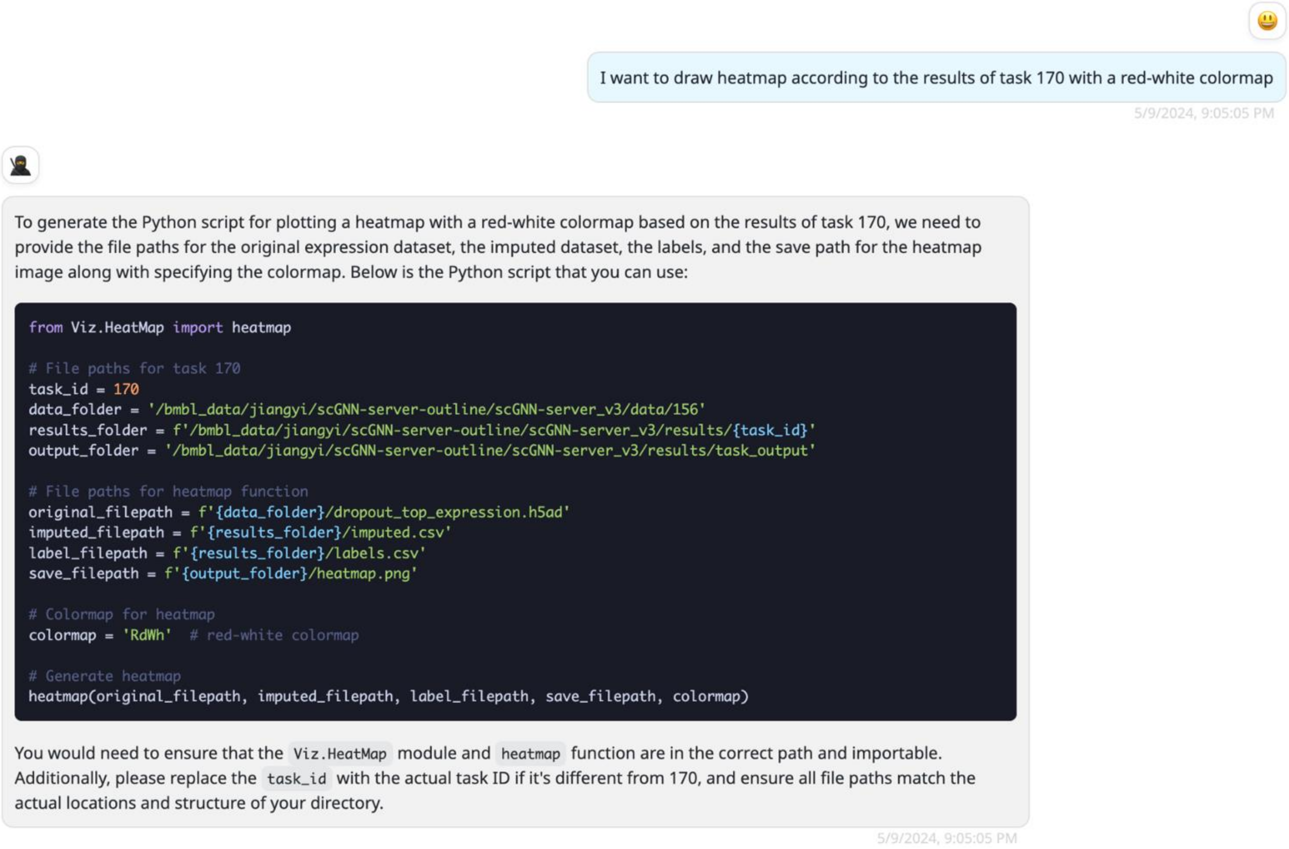
5/9/2024, 8:54:40 PM

task 171 result: 170_heatmapimputed.png

7. When figures are ready, they will be provided automatically

An example by using scGNN3.0-server

8. You can also define customized colors



I want to draw heatmap according to the results of task 170 with a red-white colormap

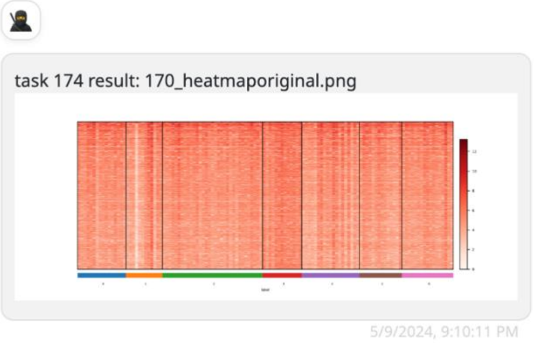
5/9/2024, 9:05:05 PM

To generate the Python script for plotting a heatmap with a red-white colormap based on the results of task 170, we need to provide the file paths for the original expression dataset, the imputed dataset, the labels, and the save path for the heatmap image along with specifying the colormap. Below is the Python script that you can use:

```python
from Viz.HeatMap import heatmap

# File paths for task 170
task_id = 170
data_folder = '/bmbl_data/jiangyi/scGNN-server-outline/scGNN-server_v3/data/156'
results_folder = f'/bmbl_data/jiangyi/scGNN-server-outline/scGNN-server_v3/results/{task_id}'
output_folder = '/bmbl_data/jiangyi/scGNN-server-outline/scGNN-server_v3/results/task_output'

# File paths for heatmap function
original_filepath = f'{data_folder}/dropout_top_expression.h5ad'
imputed_filepath = f'{results_folder}/imputed.csv'
label_filepath = f'{results_folder}/labels.csv'
save_filepath = f'{output_folder}/heatmap.png'

# Colormap for heatmap
colormap = 'RdWh'  # red-white colormap

# Generate heatmap
heatmap(original_filepath, imputed_filepath, label_filepath, save_filepath, colormap)
```

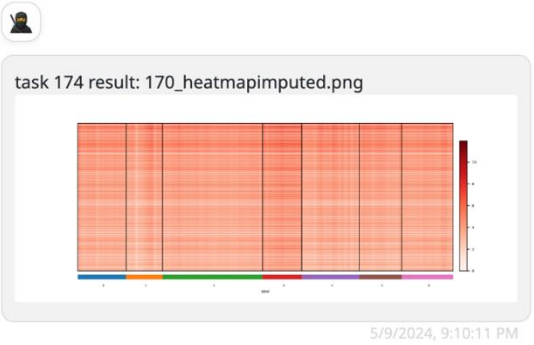
You would need to ensure that the `Viz.HeatMap` module and `heatmap` function are in the correct path and importable. Additionally, please replace the `task_id` with the actual task ID if it's different from 170, and ensure all file paths match the actual locations and structure of your directory.

5/9/2024, 9:05:05 PM



task 174 result: 170_heatmaporiginal.png

5/9/2024, 9:10:11 PM

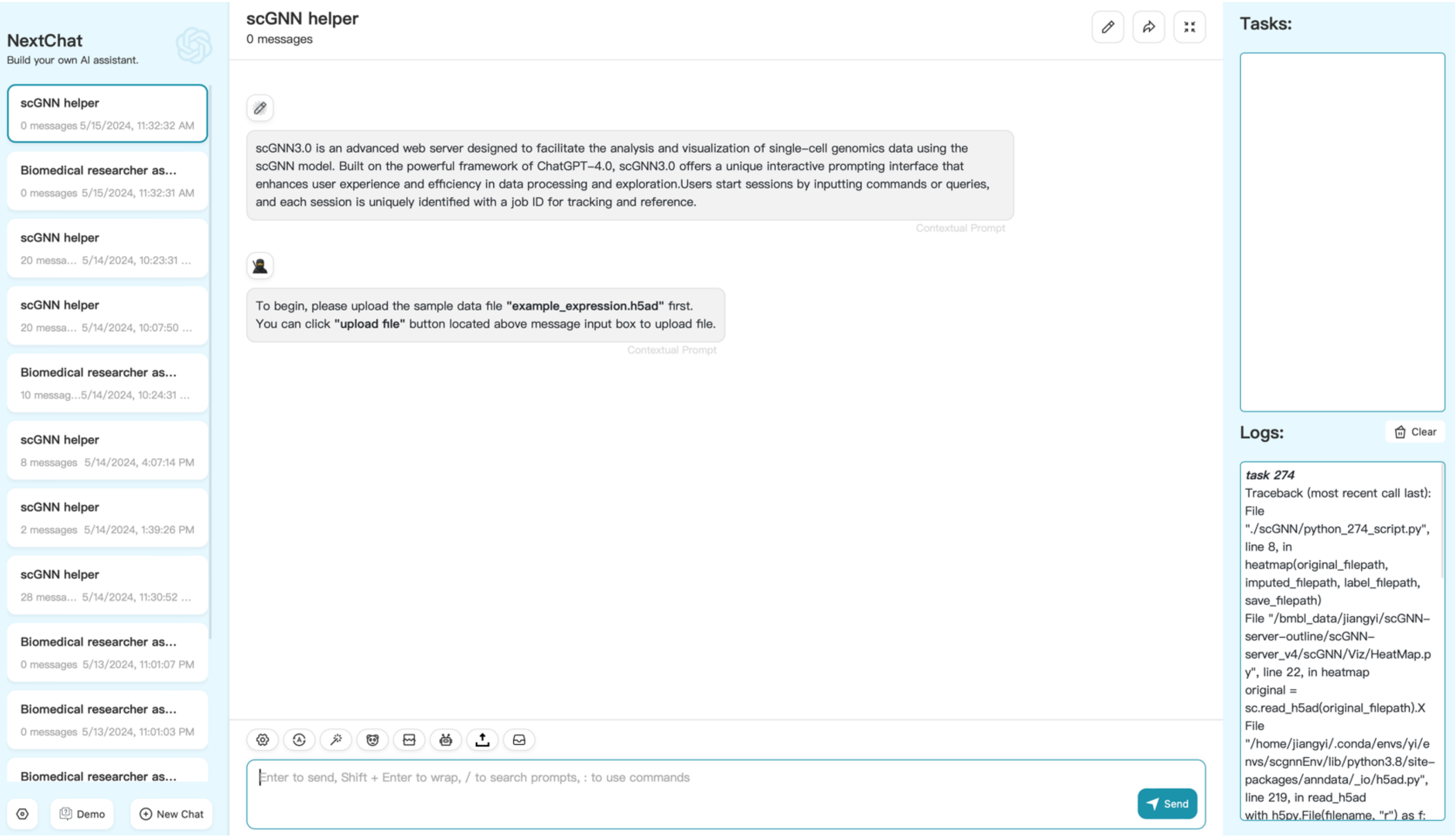


task 174 result: 170_heatmapimputed.png

5/9/2024, 9:10:11 PM

scGNN3.0-server

The server is ready!


NextChat
Build your own AI assistant.
scGNN helper
0 messages 5/15/2024, 11:32:32 AM
Biomedical researcher as...
0 messages 5/15/2024, 11:32:31 AM
scGNN helper
20 messa... 5/14/2024, 10:23:31 ...
scGNN helper
20 messa... 5/14/2024, 10:07:50 ...
Biomedical researcher as...
10 messag...5/14/2024, 10:24:31 ...
scGNN helper
8 messages 5/14/2024, 4:07:14 PM
scGNN helper
2 messages 5/14/2024, 1:39:26 PM
scGNN helper
28 messa... 5/14/2024, 11:30:52 ...
Biomedical researcher as...
0 messages 5/13/2024, 11:01:07 PM
Biomedical researcher as...
0 messages 5/13/2024, 11:01:03 PM
Biomedical researcher as...
Demo
New Chat
scGNN helper
0 messages
Contextual Prompt
Contextual Prompt
Enter to send, Shift + Enter to wrap, / to search prompts, : to use commands
Send
Tasks:
Logs:
Clear

scGNN3.0 is an advanced web server designed to facilitate the analysis and visualization of single–cell genomics data using the scGNN model. Built on the powerful framework of ChatGPT–4.0, scGNN3.0 offers a unique interactive prompting interface that enhances user experience and efficiency in data processing and exploration.Users start sessions by inputting commands or queries, and each session is uniquely identified with a job ID for tracking and reference.

To begin, please upload the sample data file **"example_expression.h5ad"** first. You can click **"upload file"** button located above message input box to upload file.

```
task 274
Traceback (most recent call last):
File
"./scGNN/python_274_script.py",
line 8, in
heatmap(original_filepath,
imputed_filepath, label_filepath,
save_filepath)
File "/bmbl_data/jiangyi/scGNN–
server–outline/scGNN–
server_v4/scGNN/Viz/HeatMap.p
y", line 22, in heatmap
original =
sc.read_h5ad(original_filepath).X
File
"/home/jiangyi/.conda/envs/yi/e
nvs/scgnnEnv/lib/python3.8/site–
packages/anndata/_io/h5ad.py",
line 219, in read_h5ad
with h5py.File(filename, "r") as f:
```

Flask PyTorch

CONDA PostgreSQL

https://bmblx.bmi.osumc.edu/scgnn3

Thank You