scGPT - user guide

Qi Liu

Data Science Group, GHDDI

qi.liu@ghddi.org

2023.08.29

Get Start

先下载 VS Code，远程连接 cluster，步骤如下：

1. 下载 VS Code: https://code.visualstudio.com/
2. 打开 VS Code，点击侧边栏的扩展
   - 搜索框中输入 ssh，下载 Remote SSH，并配置服务器[2],
   - .ssh/config 内容参考 Figure 2
3. 在 VS Code 中连接远程服务器
   - 点击侧边栏的"远程资源管理器"
   - 连接刚才配置好的服务器，选择在当前/新的窗口打开
   - 左下角显示已经连接后，选择侧边栏"打开文件夹"，打开 cluster 中想要创建项目的路径，或者已经存在的路径

后续就可以直接在 cluster 上进行开发了.

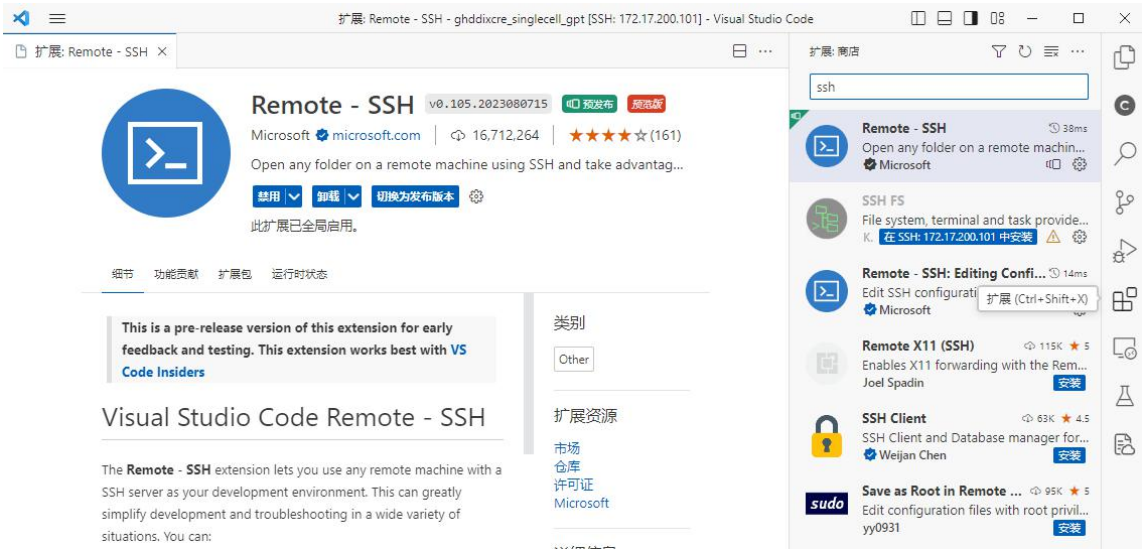
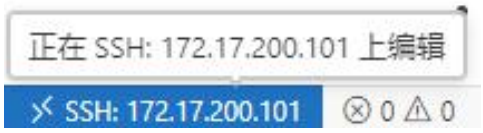


Figure 1 安装 Remote SSH 扩展

```
10  Host 172.17.200.101
11    HostName 172.17.200.101
12    User qiliu02
13    Port 22
14    IdentityFile "C:\Users\liuqi\.ssh\id_rsa"
```

Figure 2 remote ssh 的 config



(Figure 3-1)

（Figure 3-2)

Figure 3 (1）连接 server，（1）打开文件夹

Conda env

安装好 VS Code 和远程连接好 Cluster 后，下面安装 conda，并激活 scGPT 的 Conda env.

假如当前个人账号没有安装 Conda 的话，建议先安装[3]，如使用下面命令：

wget https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Linux-x86_64.sh

bash ./Anaconda3-2023.07-2-Linux-x86_64.sh

101 服务器上已经安装好了用于使用 scGPT 的环境，可选 Chen Liang (梁忱）提供的环境：

Python 解释器： /home/cliang02/work/bin/cre-python

Conda 环境：/home/cliang02/work/software/common/proglang/mambaforge/envs/cre

或者 Qi Liu (刘奇）提供的环境：

Conda 环境：/home/qiliu02/miniconda3/envs/single_cell_gpt

新建 Terminal 后，可以输入命令激活：

conda activate /home/cliang02/work/software/common/proglang/mambaforge/envs/cre

or

conda activate /home/qiliu02/miniconda3/envs/single_cell_gpt

就切换到了我们运行 scGPT 的环境了.

注意：conda env 中没有安装 scgpt package，而是在项目中源码安装 scGPT，这是为了随时保持最新和方便修改源码. 后面的预处理脚本用 sys.path.append 的方式引用了源码包，若有相关问题，可以联系 Qi Liu.

Figure 4 新建终端

```
(base) [qiliu02@comput101 ~]$ conda activate /home/cliang02/work/software/common/proglang/mambaforge/envs/cre
(cre) [qiliu02@comput101 ~]$ python --version
Python 3.10.11
(cre) [qiliu02@comput101 ~]$ conda activate /home/qiliu02/miniconda3/envs/single_cell_gpt
(single_cell_gpt) [qiliu02@comput101 ~]$ python --version
Python 3.7.13
```

Figure 5 激活 scgpt 运行的 conda 环境

Prepare data

使用我们数据预处理脚本来将.h5ad 数据处理成 scGPT pretraining 需要的 input embedding 数据集。

首先安装 git [4]，终端输入`git --version`测试，已安装的话跳过.
使用 git clone，将预处理脚本 clone 到自己的路径：
git clone https://github.com/qiliu-ghddi/singlecell_gpt

clone 后，进入到`data`路径下，将我们的要处理的数据放（或者软连接）到`data/raw`，
运行`build_large_scale_data.py`，给定`*.h5ad`文件的路径，将其用 scgpt.scbank 处理为便于大规模数据处理的格式；
接着`binning_mask_allcounts.py`，将上一步的输出，处理为能够作为 pretraining scGPT 的 embedding 数据集。

命令如下：
cd data
conda activate <env> # 参考教程里的激活 conda env，激活运行我们的 scgpt 环境
python build_large_scale_data.py --input-dir "raw/" --output-dir "./preprocessed"
python binning_mask_allcounts.py --data_source "./preprocessed/all_counts/"

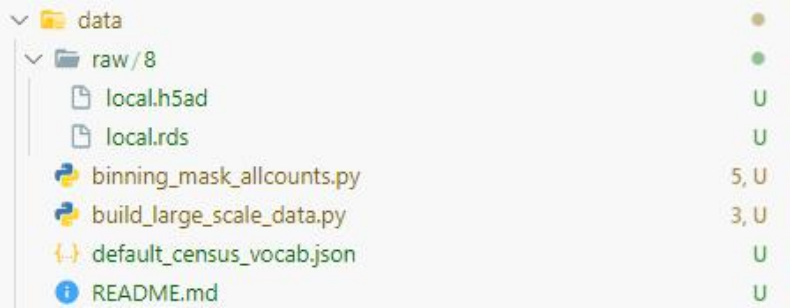


Figure 6-1

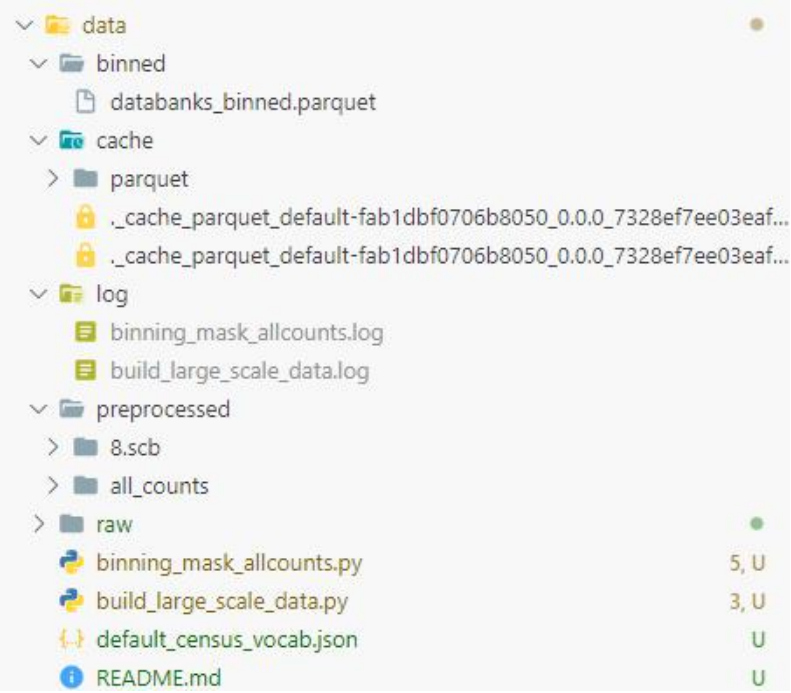


Figure 6-2

Figure 6 (6-1）处理前的数据，(6-2)处理后的数据和结果

Pretrain scGPT

我们 conda env 环境中没有直接安装 scGPT[5]， 下面源码安装 scGPT，注册 wandb[7]，随后预训练 scGPT.

clone scGPT 源，将其中的`scGPT/scgpt`源码放在我们的项目根路径下（和 code, data 平齐）.

由于 scGPT 使用了 wandb 作为日志记录的工具，所以建议去 wandb 上注册一个账号，激活后，新建终端，输入`wandb login`，输入账号和密码即可，只需要配置一次，后续就不用管了.

进入到`code`路径下，运行`python pretrain_scGPT.py`，将`data_source`配置为我们前面处理得到的`binned`数据集的路径，即可开始训练. 更多参数，请输入`python pretrain_scGPT.py --help`查看.

```
cd code
python pretrain_scGPT.py --data_source "../data/binned/"
```

训练完后，在`save/xxx-<datetime>`路径下，会保存`checkpoint`即`best_model`，及参数`args.json`和`vocab.json`文件，可以作为后面`finetune`环节的输入.

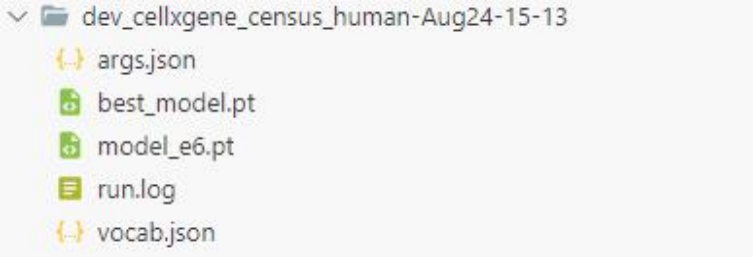


Figure 7 保存得到的 checkpoint 和结果

Down-stream tasks and evaluation

scGPT 提供了一些下游 finetune 任务的 jupyter notebooks [8]，将我们训练得到的结果，作为参数，可以运行这些下游任务.

在`examples`中提供了`finetune_integration`作为例子，修改脚本中的`load_model`的值，为我们上面训练得到的`checkpoint`所在的路径，即可运行. 其会下载 PBMC 10K`数据，并用我们得到的模型进行`batch integration`的 finetuning，最后将结果保存到`wandb`的`run`中.

```
hyperparameter_defaults = dict(
    seed=42,
    dataset_name="PBMC_10K",
    do_train=True,
    load_model="../save/dev_databanks_sample-Aug15-18-16",
    mask_ratio=0.4,
    epochs=30,
    n_bins=51,
    GEPC=True,  # Masked value prediction for cell embedding
    ecs_thres=0.8,  # Elastic cell similarity objective, 0.0 to 1.0, 0.0 to disable
    dab_weight=1.0,
    lr=1e-4,
    batch_size=64,
    layer_size=128,
    nlayers=4,
    nhead=4,
    # if load model, batch_size, layer_size, nlayers, nhead will be ignored
    dropout=0.2,
    schedule_ratio=0.9,  # ratio of epochs for learning rate schedule
    save_eval_interval=5,
    log_interval=100,
    fast_transformer=True,
    pre_norm=False,
    amp=True,  # Automatic Mixed Precision
)
run = wandb.init(
    config=hyperparameter_defaults,
    project="finetune-dev_databanks_sample-Aug15-18-16",
    reinit=True,
    settings=wandb.Settings(start_method="fork"),
)
```

Figure 8 `finetune_integration.py`中要关注的参数

参考

[1] [VS Code](https://code.visualstudio.com/)

[2] [Remote Development using SSH](https://code.visualstudio.com/docs/remote/ssh#_getting-started)

[3] [Anaconda Download](https://www.anaconda.com/download/)

[4] [Git Download](https://git-scm.com/downloads)

[5] [scGPT](https://github.com/bowang-lab/scGPT)

[6] [scGPT tutorials](https://github.com/bowang-lab/scGPT/tree/main/tutorials)

[7] [Wandb](https://wandb.ai/)

[8] [Wandb login](https://docs.wandb.ai/ref/cli/wandb-login)