Setup process for using RibbonFold:

Created by: Abhirup Paul | Newberry Lab | University of Texas, Austin
https://www.newberrylab.com
Date: June 04, 2025

Note: All the codes are to be written in Linux terminal or Ubuntu for Windows.
Github: https://github.com/Mingchenchen/RibbonFold

1. **Update and upgrade existing packages:**
    - ~$ sudo apt update && sudo apt upgrade -y

2. **Installing packages:**
    - ~$ sudo apt install -y python3 python3-pip python3-venv git wget
      #### python3 and python3-pip are needed to install further dependencies. Python3-venv is needed for created isolated environments.

3. **Clone the RibbonFold repository from Github:**
    - ~$ ls
    - ~$ cd #directory of choice
    - ~$ git clone https://github.com/Mingchenchen/RibbonFold.git
    - ~$ cd RibbonFold

4. **Creating a virtual environment for RibbonFold:**
    - ~$ conda create -n ribbon_env python=3.9
      #### This is a conda based setup that requires NVIDIA GPU support.
      #### If running only on CPU, change the following:
      ~$ pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu
      ~$ pip install tensorflow-cpu==2.6.0
      Then, modify inference.py - device = torch.device("cpu")
    - ~$ conda activate ribbon_env
    - ~$ conda deactivate *## if you want to deactivate the env*

5. **Installing CUDA and Python dependencies:**
    - ~$ conda install -y cudatoolkit=11.8 -c nvidia
    - ~$ pip install torch==2.1.0+cu118 torchvision==0.16.0+cu118 torchaudio==2.1.0+cu118 -f https://download.pytorch.org/whl/torch_stable.html
    - ~$ ls
    - ~$ pip install torchtyping==0.1.4 functorch tensorflow-cpu==2.6.0 tensorflow-estimator==2.14.0
    - ~$ pip install pandas==1.3.5 scipy==1.5.4 biopython dm-tree treelib tqdm ml_collections pytz python-dateutil contextlib2 PyYAML --no-deps
    - ~$ pip install protobuf==3.19.6

6. **Download the model weights:**
    - ~$ wget https://zenodo.org/records/15128410/files/model_checkpoints.tar.gz?download=1
      ##### Rename the file to "model_checkpoints.tar.gz"
    - ~$ mkdir -p ./ckpt
    - ~$ tar -xzvf model_checkpoints.tar.gz -C ./ckpt

7. Running RibbonFold:

Prepare the MSA feature file:

- You'll have to make your own MSA file in A3M format (.a3m) from AlphaFold2. You also have to clean and process the MSA file for process_msa_file.py to run. Here's a detailed instruction on how to do so:
    a. ***Generate the structure module using AlphaFold2. Parameters used:***
        - *num relax: 5*
        - *template mode: pdb100*
        - *msa_mode: mmseqs2_uniref_env*
        - *pair_mode: unpaired_paired*
        - *num_recycles: 48 (can lower as well)*
        - *dpi: 600 (if you want to save the structures)*

    b. ***Download the results***

    c. ***Extract the msa file (.A3M) file and run a custom python code, "clean_msa.py"***
        - *This code trims the msa file to a accessible file for RibbonFold. You can find this code in my Signals LNB or Githib (https://github.com/APaul26).*
- Once done, run these commands. Here, I'm using the example provided (5oqv – Aß(1-42))
- ~$ python process_msa_file.py --input_fasta ./examples/5oqv.fasta --msa_file ./examples/5oqv_msa.a3m --output ./examples/5oqv_msa.pkl.gz
  ##### this preprocess the MSA features from an AlphaFold2 msa file. A pkl.gz file will be generated, and this file should be passed through the following inference script.
  ##### update the locations of the fasta file and alignment file based on your preference

Run inference.py:

- ***Sample code, modify the script and run:***

```
~$ CHECKPOINT_PATH="./ckpt/model_ckpt_001.pt"
INPUT_PKL_FILE="./examples/5oqv_msa.pkl.gz"
OUTPUT_DIR="./results/"
ROUNDS=10
python inference.py \
--checkpoint ${CHECKPOINT_PATH} \
--input_pkl ${INPUT_PKL_FILE} \
--ribbon_name 5oqv \
--output_dir ${OUTPUT_DIR} \
--rounds ${ROUNDS} \
--use_dropout true \
--use_init_structure true
```

- ***Test code for alpha_synuclein:*** *### changed the no. of chains to 3 to account for memory*

```
~$ CHECKPOINT_PATH="./ckpt/model_ckpt_001.pt"
INPUT_PKL_FILE="./test/SYUA_msa.pkl.gz"
OUTPUT_DIR="./test/"
ROUNDS=3
python inference_test.py \  ### Custom modified code
--checkpoint ${CHECKPOINT_PATH} \
--input_pkl ${INPUT_PKL_FILE} \
--ribbon_name SYUA \
--output_dir ${OUTPUT_DIR} \
--rounds ${ROUNDS} \
--use_dropout true \
--use_init_structure true
```