**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

1. **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.*

1. **Clone the RibbonFold repository from Github:**

* ~$ ls
* ~$ cd #directory of choice
* ~$ git clone https://github.com/Mingchenchen/RibbonFold.git
* ~$ cd RibbonFold

1. **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*

1. **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

1. **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

1. **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:

1. ***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)*

1. ***Download the results***
2. ***Extract the msa file (.A3M) file and run a custom python code, “clean\_msa.py”***

* *First convert the file to a .fasta file and the run the code.*
* *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*](https://github.com/APaul26)*).*
* *After the file has been trimmed, convert it back to .a3m file.*
* 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
* python process\_msa\_file.py --input\_fasta ./test/trunc\_SYUA.fasta --msa\_file ./test/trunc\_SYUA\_msa.a3m --output ./test/trunc\_SYUA\_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:***

~$ CHECKPOINT\_PATH="./ckpt/model\_ckpt\_001.pt"

INPUT\_PKL\_FILE="./test/trunc\_SYUA\_msa.pkl.gz"

OUTPUT\_DIR="./test/"

ROUNDS=5

python inference\_test.py \

--checkpoint ${CHECKPOINT\_PATH} \

--input\_pkl ${INPUT\_PKL\_FILE} \

--ribbon\_name trunc\_SYUA \

--output\_dir ${OUTPUT\_DIR} \

--rounds ${ROUNDS} \

--use\_dropout true \

--use\_init\_structure true