RoseTTAFold and Protein Structure Prediction

CCATS Group



RoseTTAFold Workflow

- 1. Input sequence
 - · Amino Acid Sequence in FASTA format (.fa)
- 2. RoseTTAFold Program
 - · Conda Environments (GPU, Folding; ~3.8 Gb)
 - RoseTTAFold Software (~6.9 Gb)
 - PyRoseTTA License (https://els2.comotion.uw.edu/product/pyrosetta)
- 3. RoseTTAFold Databases (~460 Gb)
 - Uniref30, Reduced BFD/Mgnify, Structure Templates (RCSB)

If all goes well, you get 5 predicted monomer structures

03_rosettafold

On OSCER, copy this directory to your home:

cp /home/van/MacroMol_Modelling_Tutorial/03_rosettafold.

- The folder is organized as follow:
- > 03_rosettafold
 - > Results
 - > Tutorial
 - > 00_monomer_pred
 - > 01_complex_pred

03_rosettafold/Tutorial/00_monomer_pred

- This folder contains:
 - bax.fa (Bax)
 - rf_monomer_pred.slurm
 - SLURM script to run RoseTTAFold
 - To run:

sbatch rf_monomer_pred.slurm

bax.fa (Bax FASTA sequence)

```
1 >bax
2 MDGSGEQPRGGGPTSSEQIMKTGALLLQGFIQDRAGRMGGEAPE
    LALDPVPQDASTKKLSECLKRIGDELDSNMELQRMIAAVDTDSP
    REVFFRVAADMFSDGNFNWGRVVALFYFASKLVLKALCTKVPEL
    IRTIMGWTLDFLRERLLGWIQDQGGWDGLLSYFGTPTWQTVTIF
    VAGVLTASLTIWKKMG
```

rf_monomer_pred.slurm

```
/bin/bash
 2 #SBATCH --partition=gpu
 3 #SBATCH --output=%j.out
                             SLURM Allocation requests
 4 #SBATCH --error=%j.err
 5 #SBATCH --time=48:00:00
 6 #SBATCH --job-name=rosetta
 8 date
                             Load CUDA library (for GPU)
10 module load CUDA/11.3.1
11 ln -sf /home/van/Programs/RoseTTAFold/run_pyrosetta_ver.sh
   ./run_pyrosetta_ver.sh bax.fa .
15 date
                              Create link of RoseTTAFold Script and run
                              the program taking the FASTA sequence
                              as input 1 and "." as the working directory
```

run_pyrosetta.sh (made by Van)

RF run_pyrosetta.sh [input FASTA] [working directory] RoseTTAFold conda activate Prepare MSA **RF** make_msa.sh [input FASTA] > [input.a3m] conda activate folding Predict Secondary Structure **RF** make_ss.sh [input.a3m] > [input.ss2] Search Templates Combine [input.a3m] [input.ss2] > [input.msa0.ss2.a3m] HHsearch [input.msa0.ss2.a3m] > [input.hhr] & [input.atab] Predict Distance & Orientation **RF** predict_pyrosetta.py \ [input.a3m] [input.hhr] [input.a3m] > [input.3track]

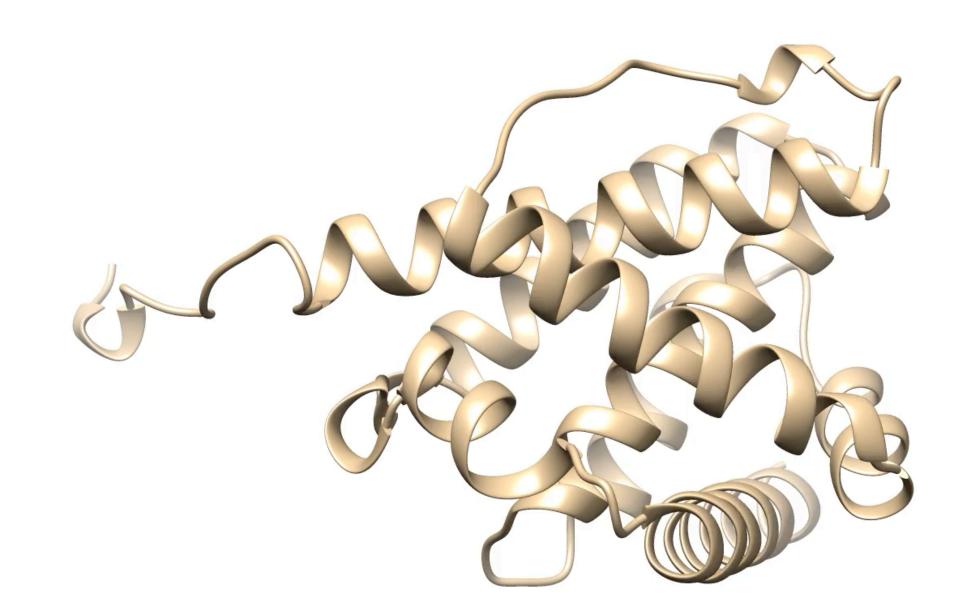
Monomer Prediction **RF** RosettaTR.py \ [input FASTA] [input.3track > [input.pdb] Error Estimation (DeepAccNet-MSA) **RF** ErrorPredictorMSA.py \ [input.3track] > [input.crderr.pdb] 5 Predicted Models (with separate PDB for error)

[name] : name of input (auto) NOTE: Bold : program Italic : scripts

RF: RoseTTAFold

Visualizing Model 1 Error on UCSF Chimera

- Download the models folder to your local computer
- Load model_1.crderr.pdb on Chimera
- Change atom coloring to error (found in B Factor column)
 - Select > Structure > protein
 - Action > Atom/bonds > Show
 - Tools > Depiction > Render by Attribute



03_rosettafold/Tutorial/01_complex_pred.slurm

- This folder contains:
 - paparde.fa (from Alphafold Tutorial)
- Need to make two separate FASTA files:

head -n 2 paparde.fasta > pard.fa

tail -n 2 paparde.fasta > pare.fa

- rf_complex_pred.slurm
 - SLURM script to run RoseTTAFold

To Run:

sbatch rf_complex.slurm

paparde.fa (from Alphafold

- 1 > ParD
- 2 GSHMMSLKWTRKAAADLDAIYDHYVVLIGPEKALKAVQDIVEQVKPLQQ VANQGAGRPSEVPGVRTLTLERWPFSAPFRVKGKEIQILRIDRVEITP
- 3 >ParE
- 4 GSHMMSTVVSFRADDALVAALDELARATHRDRPYHLRQALAQYLERQQW QVAAIDEGLADANAGRLLEHIEIEKRWGLQ

rf_complex_pred.slurm (made by Van)

(van) conda activate RoseTTAFold OPTIONAL ----Copy from RoseTTAFold (RF): make_joint_MSA_bacterial.py make_msa.py (2track) predict_complex.py Prepare MSA for each subunit RF make_msa.sh\ [subunit1 FASTA] > [subunit1.a3m] **RF** make msa.sh\ [subunit2 FASTA] > [subunit2.a3m] Make Pair Alignments **RF** make_joint_MSA_bacterial.py \ [subunit1.a3m] [subunit2.a3m] > paired.a3m

Complex Template Prediction **RF** predict_msa.py paired.a3m > complex.npz

conda activate

folding

Complex Prediction

RF predict_complex.py\

filtered.a3m > model_#_SML_P.pdb> complex.pdb



Generate Sidechains and Error Prediction

RF RosettaTR.py\

complex.pdb [input FASTA] [input.3track] > [output.pdb]

Filter Pair Alignment

HHfilter paired.a3m > filtered.a3m

NOTE: bold-underline : constant file name