ROSETTA Folding Tutorial – Step- by-step Instructions

*** BLUE text means that these files and/or this information is provided.**
*** RED text means that this material will NOT be conducted during the workshop**
*** If you want to try making files that already exist (e.g., input files), write them to a different directory!**

1. Prepare your input files

a. FASTA file of your sequence

- Get sequence in FASTA format from NCBI
  - The `2LZM_.fasta` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax` directory
  - Go to http://www.ncbi.nlm.nih.gov/protein/.
  - Type in 2LZM in the search bar at top.
  - Click link called 2LZM to see obsolete version.
  - Copy all the sequence information, including the line beginning with “>” into a file called `2LZM_.fasta`.
  - Open `2LZM_.fasta` in a text editor and remove first 57 residues of the sequence so that it begins with the sequence ITKDE and save the file.
  - Move `2LZM_.fasta` to the `$WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax` directory

b. Prepare PDB of native structure (optional, covered by Steven Combs)

i. The `2LZM_.pdb` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax` directory

ii. – Search for and download the 2LZM PDB file from http://www.pdb.org/pdb/home/home.do. Save this file to 2lzm.pdb

iii. Remove first 57 residues from coordinates Step so that it begins with the sequence ITKDE

iv. `python $ROSETTA_SCRIPTS/clean_pdb.py 2lzm.pdb nochain`

v. `mv 2lzm_nochain.pdb $WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax/2LZM_.pdb`

c. 3mer and 9mer fragment libraries

i. The `2LZM` fragment files are already provided for you in the `$WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax` directory (`aa2LZM_03_05.200_v1_3 and aa2LZM_09_05.200_v1_3`)

- **Using Robetta** (for the purposes of this workshop)
- The 3mer and 9mer fragment files are provided for you in the `$WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax` directory
- If you are an academic or non-profit user of ROSETTA, make sure you’re registered at http://robetta.bakerlab.org/
- Under “Services,” click “submit” under “Fragment Libraries”

- Fill in the form; copy/paste all the text in `2LZM_.fasta` into the provided field. Under "Target name" put 2LZM_. **Note:** If you are benchmarking, would want to exclude homologues.
- Click "Submit." You can see your position in the queue by clicking "Queue" under "Fragment Libraries." This should not take very long.
- Your fragment files should be called `aa2LZM_03_05.200_v1_3` and `aa2LZM_09_05.200_v1_3`. Save all the files to `$WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax`

ii. Using `make_fragments.pl`

- **We will not run `make_fragments.pl` during the workshop!**
- If you are working for a for-profit institution, will need to use the `make_fragments.pl` script in `your_rosetta_directory/rosetta-3.2/rosetta_fragments`.
- In order to use it, will first need to install PSIBLAST, the non-redundant (NR) database, and perhaps PSIPRED
- Will need to modify `make_fragments.pl` in order to reflect the paths specific to your case (will not do during workshop)
- For a usage statement, run: `your_rosetta_directory/rosetta-3.2/rosetta_fragments/make_fragments.pl`

d. Options file

- The `2LZM_abrlx.options` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/folding/1-input_AbinitioRelax` directory
- Try to make one on your own. ROSETTA ignores lines beginning with # (these are comments.)
- Replace variable names, such as `$ROSETTA_DATABASE` and `$WORKSHOP_ROOT` with your specific absolute paths.

```
- $ROSETTA_SCRIPTS/replace_env_variables.py
$WORKSHOP_ROOT/tutorials/folding/1-
input_AbinitioRelax/2LZM_abrlx.options
```

- Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated; covered by Steven Combs).

2. Run ROSETTA AbinitioRelax application

a. Make sure all the filenames and paths in the options file are correct!

b. Go to the folding tutorial main directory

c. Type the following command line. It is also found in the `command` file in `$WORKSHOP_ROOT/tutorials/folding/2-command_AbinitioRelax`

```
- $ROSETTA_BIN/AbinitioRelax.$ROSETTA_SUFFIX
@$WORKSHOP_ROOT/tutorials/folding/1-
input_AbinitioRelax/2LZM_abrlx.options -database
$ROSETTA_DATABASE >& $WORKSHOP_ROOT/ tutorials/folding/3-
analyze_AbinitioRelax/abrlx.log &
```

3. Analyze your data

a. Score and extract PDBs

i. The silent file and PDB files of the lowest-scoring models are provided for you in the `$WORKSHOP_ROOT/tutorials/folding/3-analyze_AbinitioRelax/example_output` directory

ii. If you ran more than one job, you will need to combine silent files into one file.

```
- cd $WORKSHOP_ROOT/tutorials/folding/3-
analyze_AbinitioRelax/
- $ROSETTA_BIN/combine_silent.$ROSETTA_SUFFIX –
database $ROSETTA_DATABASE -in:file:silent 2LZM_0*.out
-in:file:silent_struct_type binary –in:file:fullatom -
out:output –out:file:silent 2LZM_all_models_silent.out
–out:file:silent_struct_type binary –out:file:fullatom
```

iii. `$ROSETTA_SCRIPTS/score_scatter_plot.py –h`

```
- $ROSETTA_SCRIPTS/score_scatter_plot.py --x_axis=rms
--y_axis=score --silent=2LZM_all_models_silent.out
2LZM_models.table >& score.log &
```

iv. The `2LZM_models.table` and other files are provided for you in the `$WORKSHOP_ROOT/tutorials/folding/3-analyze_AbisnitioRelax/example_analysis` directory

- If you're not already there, cd into `$WORKSHOP_ROOT/tutorials/folding/3-analyze_AbisnitioRelax`
- Sort the `2LZM_models.table` by the score column from lowest -> highest.
  - `sort –nk3 2LZM_models.table > 2LZM_models_sorted.table`
- Take the top 5-10 models by score to look at
  - `head –n 10 2LZM_models_sorted.table`
- Can also sort by RMSD (`sort –nk2`)

v. Now you know the tags of the models you want to extract from the binary silent file, which you can do with the following command line:

```
- $ROSETTA_BIN/score_jd2.$ROSETTA_SUFFIX –database
$ROSETTA_DATABASE -in:file:silent
2LZM_all_models_silent.out -in:file:silent_struct_type
binary –in:file:fullatom -out:output -out:pdb -
out:file:fullatom -in:file:tags S_00000175_3
S_00000129_1 S_00000026_2 S_00000168_2 S_00000028_4
```

b. Score vs. RMSD plots

i. Assume that your lowest-scoring model is the native. Determine which model is the lowest-scoring.

```
- grep SCORE 2LZM_all_models_silent.out | sort –nk2 |
head –n 5
```

- The tag of the correct model is in the very last field (column) of the first line.

ii. Rescore the models, computing RMSD to the lowest-scoring model.

```
- $ROSETTA_BIN/score_jd2.$ROSETTA_SUFFIX -database
$ROSETTA_DATABASE -in:file:native
S_00000175_3_0001.pdb -in:file:silent
2LZM_all_models_silent.out –in:file:silent_struct_type
binary –in:file:fullatom –out:file:silent
```

```
2LZM_all_models_rescored_silent.out –
out:file:silent_struct_type binary –out:file:fullatom
```

iii. Make a table of the scores and RMSDs of your models:

```
- $ROSETTA_SCRIPTS/scripts/score_scatter_plot.py --
x_axis=rms --y_axis=score --silent
2LZM_all_models_rescored_silent.out
2LZM_score_vs_rmsd.table
```

iv. Make a scatter plot using the program of your choosing. For example, read in the `2LZM_score_vs_rmsd.table` file into Excel, and make an X-Y scatter plot.

c. Look at best-scoring models by opening them in PyMol or the molecular graphics program of your choosing

4. Preparation for running MembraneAbinitio

a. Spanfile

i. The `BRD4.span` spanfile is already provided in the `$WORKSHOP_ROOT/tutorial/folding/4-input_MembraneAbinitio` directory

ii. Generate the FASTA file for your α-helical membrane protein as in Step 1.a > `$WORKSHOP_ROOT/tutorial/folding/4-input_MembraneAbinitio/BRD4_.fasta`

iii. Go to http://octopus.cbr.su.se/

iv. Copy the BRD4 sequence (only the sequence!) into the provided box and click “submit”

v. After a few minutes, it should generate a text file. Click on the OCTOPUS topology file link. Copy the contents of this file into a file called `BRD4.octopus` and move this file to `$WORKSHOP_ROOT/tutorial/folding/4-input_MembraneAbinitio`

vi. Now run `$ROSETTA_SCRIPTS/octopus2span.pl BRD4.octopus > BRD4.span`

- This will create a `BRD4.span` spanfile describing the membrane-spanning regions of BRD4.

b. LIPS file

- The `BRD4.lips4` LIPS file is already provided in the `$WORKSHOP_ROOT/tutorials/folding/4-input_MembraneAbinitio` directory
- **Note**: Can only generate this file if have BLAST and NR database installed. **We will not run this script during the workshop!**
- **Example:**

```
your_rosetta_directory/rosetta-
3.2/rosetta_source/src/apps/public/membrane_abinitio/run_l
ips.pl BRD4.fasta BRD4.span  /blast/bin/blastpgp
/nr_database your_rosetta_directory/rosetta-
3.2/rosetta_source/src/apps/public/membrane_abinitio/align
blast.pl
```

c. 3mer and 9mer fragment libraries

i. The BRD4 fragment files are already provided in the `$WORKSHOP_ROOT/tutorials/folding/4-input_MembraneAbinitio` directory (`aaBRD4_03_05.200_v1_3` and `aaBRD4_09_05.200_v1_3`)

ii. See Step 1.c.i to make fragments using Robetta. **We will not run `make_fragments.pl` during the workshop!**

- FYI: In order to run with SAM secondary structure prediction as recommended, will need to run SAM independently first. Make sure SAM output is in the correct format:

```
 Pos   AA       E        H        L
 10N   1S       5N       5N       5N
  0     M     0.126    0.091    0.783
  1     N     0.126    0.053    0.822
  2     G     0.104    0.032    0.865
```

iii. Using `make_fragments.pl`

```
- your_rosetta_directory/rosetta-3.2/rosetta_fragments
make_fragments.pl —nosam —nopsipred —nojufo —samfile
BRD4.rdb —id BRD4_ BRD4.fasta >& make_fragments.log &
```

d. Options file

- The `BRD4_mem_abrlx.options` file is already provided in the `$WORKSHOP_ROOT/tutorials/folding/4-input_MembraneAbinitio` directory. Note the difference in format. There are multiple ways to pass options to ROSETTA.

- There are only a few differences between this options file and the one in Step 1.d that have to do with membrane protein-specific options

- Try to make one on your own. ROSETTA ignores lines beginning with # (these are comments.)

- Replace variable names, such as `$ROSETTA_DATABASE` and `$WORKSHOP_ROOT` with your specific absolute paths.

```
- $ROSETTA_SCRIPTS/replace_env_variables.py
$WORKSHOP_ROOT/tutorials/folding/4-
input_MembraneAbinitio/BRD4_mem_abrlx.options
```

- Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated; covered by Steven Combs).

5. Running MembraneAbinitio application

a. Make sure all the filenames and paths in the options file are correct!

b. Go to the folding tutorial main directory

c. Run the following command line. It is also found in the `command` file in `$WORKSHOP_ROOT/tutorials/folding/5-command_MembraneAbinitio`

```
- $ROSETTA_BIN/membrane_abinitio2.$ROSETTA_SUFFIX
@$WORKSHOP_ROOT/tutorials/folding/4-
input_MembraneAbinitio/BRD4_mem_abrlx.options —database
$ROSETTA_DATABASE >& $WORKSHOP_ROOT/tutorials/folding/6-
analyze_MembraneAbinitio/membrane_abinitio.log &
```

6. Analyzing MembraneAbinitio output

- The example output is already provided in the `$WORKSHOP_ROOT/tutorials/folding/6-analyze_MembraneAbinitio/example_output` directory

a. Combing and extracting silent files and extracting PDBs of membrane proteins is slightly different than for soluble proteins but is a very similar process as that described in Step 3.a

i. cd into `$WORKSHOP_ROOT/tutorials/folding/6-analyze_MembraneAbinitio`

ii. To combine silent files:

```
- $ROSETTA_BIN/combine_silent.$ROSETTA_SUFFIX -
database $ROSETTA_DATABASE -in:file:silent
BRD4_abrlx_0*.out -in:file:silent_struct_type binary -
in:file:residue_type_set centroid -in:file:spanfile
../4-input_MembraneAbinitio/BRD4.span -score:weights
$ROSETTA_DATABASE/scoring/weights/score_membrane.wts -
out:file:silent BRD4_mem_abrlx_all.out -
out:file:silent_struct_type binary -
out:file:residue_type_set centroid
```

ii. Find the lowest-scoring models:

```
- grep SCORE BRD4_mem_abrlx_all.out | sort -nk2 | head
| awk '{print $26}' > top10score.ls
- cat top10score.ls
```

iii. To extract PDBs:

```
- $ROSETTA_BIN/score_jd2.$ROSETTA_SUFFIX –database
$ROSETTA_DATABASE -in:file:silent
BRD4_mem_abrlx_all.out -in:file:tags S_00000058_1
S_00000163_2 S_00000179_1  S_00000087_3 S_00000035_2 -
in:file:silent_struct_type binary -
in:file:residue_type_set centroid –in:file:spanfile
../4-input_MembraneAbinitio/BRD4.span –score:weights
$ROSETTA_DATABASE/scoring/weights/score_membrane.wts -
out:pdb -out:file:residue_type_set centroid
```

b. See Step 3.b concerning score vs. RMSD plots etc. If you want to rescore your models and compute the RMSD against the lowest-scoring model, repeat Step 6.a.2, and add the following options: `-in:file:native S_00000058_1_0001.pdb -evaluation: rmsd NATIVE TM $WORKSHOP_ROOT/tutorials/folding/4-input_MembraneAbinitio/TM_rms.txt`

- `TM_rms.txt` is a file containing the residues over which you want to compute the CA-RMSD (the membrane-spanning regions in this case). It has the format:

```
RIGID 6 26
RIGID 31 51
RIGID 58 78
RIGID 97 117
```

7. Preparation for folding with restraints

a. Constraints file

- The `2LZM_dist_w1.cst` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/folding/7-input_FoldConstraints` directory
- The cst file has the basic format:

#cst type	atomtype	res#	atomtype	res#	function	EPR potential	exp_dist	weight	bin_size
AtomPair	CB	32	CB	36	SPLINE	EPR_DISTANCE	16.0	1.0	0.5
AtomPair	CB	59	CB	74	SPLINE	EPR_DISTANCE	19.0	1.0	0.5
AtomPair	CB	62	CB	71	SPLINE	EPR_DISTANCE	19.0	1.0	0.5
AtomPair	CB	62	CB	74	SPLINE	EPR_DISTANCE	25.0	1.0	0.5
AtomPair	CB	63	CB	74	SPLINE	EPR_DISTANCE	14.0	1.0	0.5

b. Options file

- The `2LZM_abrlx_cst.options` file is already provided for you in the `$WORKSHOP_ROOT/tutorials/folding/7-input_FoldConstraints` directory

- There are only a few differences between this options file and the one in Step 1.d that have to do with FoldConstraint options

- Try to make one on your own. ROSETTA ignores lines beginning with # (these are comments.)

- Replace variable names, such as `$ROSETTA_DATABASE` and `$WORKSHOP_ROOT` with your specific absolute paths.

```
- $ROSETTA_SCRIPTS/replace_env_variables.py
$WORKSHOP_ROOT/tutorials/folding/7-
input_FoldConstraints/2LZM_abrlx_cst.options
```

- Avoid mixing tabs and spaces. Be consistent in your formatting (tab-delimited or colon-separated; covered by Steven Combs).

8. Running AbinitioRelax (or MembraneAbinitio) application with restraints

- The command line file is found in the `command` file in `$WORKSHOP_ROOT/tutorials/folding/8-command_FoldConstraints.`

a. Go to the folding tutorial main directory

b. See Step 2 or Step 5 (Step 2.c for the workshop). Run the same command line, replacing the path and name of the original options file for `$WORKSHOP_ROOT/tutorials/folding/7-input_FoldConstraints/2LZM_abrlx_cst.options`

9. Analyzing folding with restraints output

- Example output files are provided for you in `$WORKSHOP_ROOT/tutorials/folding/9-analyze_FoldConstraints/example_output`

a. See Step 3.b concerning score vs. RMSD plots etc.

b. Analyze how models satisfy restraints

- An example restraint violation analysis file is provided for you in `$WORKSHOP_ROOT/tutorials/folding/9-analyze_FoldConstraints/example_analysis` (`2LZM_cst_viol.txt`).

i. cd into `$WORKSHOP_ROOT/tutorials/folding/9-analyze_FoldConstraints`

ii. You can sort the models by atom_pair_constraint score and see which models satisfy the restraints the best. For example, for 25 restraints weighted by a factor of 4 and scored with the ROSETTAEPR knowledge-based potential, the best score (100% of restraints satisfied) is –100.00 REU. Often want to filter models by some combination of total score and restraint score (see Hirst *et al*., *J. Struct. Biol.* 2011).

```
- grep SCORE 2LZM_all_models_cst_silent.out | sort -
nk17 | head | awk '{print($1"\t"$2"\t"$17"\t"$38)}' >
top10_atom_pair_constraint_score.txt
```

iii. Can also see how much restraints are violated in terms of distance. In a tcsh shell, do:

```
- ls *.pdb > pdbs.ls
- foreach pdb (`cat pdbs.ls`)
foreach?  $ROSETTA_SCRIPTS/calc_exp_viol.pl $pdb
$WORKSHOP_ROOT/tutorials/folding/7-
input_FoldConstraints/2LZM_dist_w1.cst 25 >>
2LZM_cst_viol.txt
foreach? end
```

- your `2LZM_cst_viol.txt` will look like:

```
# pdb file name       total # viol. In pdb  total sum viol (Å)   single max viol. (Å)
S_00000010_4_0001.pdb number_violations: 5 sum_violations: 11.276 max_violation: 5.389
S_00000019_3_0001.pdb number_violations: 6 sum_violations: 12.354 max_violation: 5.818
S_00000045_4_0001.pdb number_violations: 5 sum_violations: 11.376 max_violation: 4.918
S_00000066_2_0001.pdb number_violations: 6 sum_violations: 15.155 max_violation: 5.411
S_00000080_4_0001.pdb number_violations: 5 sum_violations: 12.358 max_violation: 4.373
S_00000084_3_0001.pdb number_violations: 5 sum_violations: 13.174 max_violation: 5.490
S_00000094_4_0001.pdb number_violations: 5 sum_violations: 10.863 max_violation: 5.726
S_00000115_3_0001.pdb number_violations: 5 sum_violations: 11.715 max_violation: 4.789
S_00000116_3_0001.pdb number_violations: 6 sum_violations: 12.957 max_violation: 5.384
S_00000132_0001.pdb number_violations: 6 sum_violations: 14.738 max_violation: 5.982
```