Rosetta Basics:
IO and Navigating Rosetta

Shannon Smith
shannon.t.smith.1@vanderbilt.edu
Graduate Student
Meiler Lab, Vanderbilt University
Goals for this Talk

1. General Rosetta Concepts:
   - How do I run basic Rosetta applications?
   - Input/Output: file types, options, etc.

2. Learn where things are in Rosetta
   - Your working directory is independent of these Rosetta directories
     - (AKA your data is stored outside of Rosetta)
   - Highly encourage everybody to follow along from your command line
   - Full path = path to file from home directory
     - (~ = /usr/people/molgraph/)
   - Relative path = path to file from current working directory

• Please stop me for questions!
How do we get Rosetta?

- [https://www.rosettacommons.org/software/license-and-download](https://www.rosettacommons.org/software/license-and-download)

- Weekly Releases: (e.g. “2016.37”)
  - Latest version of the code, released roughly every week
  - Every revision passes scientific performance tests

- Numbered Releases (e.g. “3.9”)
  - A weekly release that’s relabeled, released roughly every 6 months

- All tutorials use version 2017.36.59679 (May 2018)

- Links to documentation, forum and demos:
  - [https://www.rosettacommons.org/docs/latest/Home](https://www.rosettacommons.org/docs/latest/Home)
  - [https://www.rosettacommons.org/demos/latest/Home](https://www.rosettacommons.org/demos/latest/Home)
cd ~/rosetta_workshop/tutorials/short_talks/RosettaI/O/

- Where you will find all the files associated with this talk
- input_files/
- output_files/
- scripts/

- Notice these files are located outside of Rosetta (aka you do NOT want to store your input/output files in directories within ~/rosetta_workshop/rosetta/*) !!

Tip: Open each file in your favorite text editor (gedit, vi, emacs, etc.) as we introduce them throughout the talk
Basic Pipeline for Running a Rosetta Application

Input Files
- Structure
- Sequence
- Application-Specific

Output Files
- Structures
- Scores
- Log/Tracer

Rosetta Application
Structure Files: PDB

- PDBs are broken down by atom lines
- Each atom is characterized by the following (left to right across each row)
  - Atom, atom number, atom name, residue name, chain identifier, residue sequence number, x, y, z orthogonal coordinates, occupancy, temperature factor, element symbol, atom charge.
- Note: CIF not supported

Snippet from 1qys.pdb

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>419 ATOM 57 N ILE A 10</td>
<td>3.073</td>
<td>-2.269</td>
<td>16.587</td>
<td>1.00</td>
<td>40.88</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>420 ATOM 58 CA ILE A 10</td>
<td>3.976</td>
<td>-3.337</td>
<td>16.187</td>
<td>1.00</td>
<td>40.88</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>421 ATOM 59 C ILE A 10</td>
<td>3.531</td>
<td>-4.697</td>
<td>16.730</td>
<td>1.00</td>
<td>40.88</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>422 ATOM 60 O ILE A 10</td>
<td>3.465</td>
<td>-4.900</td>
<td>17.944</td>
<td>1.00</td>
<td>70.73</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>423 ATOM 61 CB ILE A 10</td>
<td>5.399</td>
<td>-3.068</td>
<td>16.702</td>
<td>1.00</td>
<td>61.80</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>424 ATOM 62 CG1 ILE A 10</td>
<td>5.870</td>
<td>-1.693</td>
<td>16.253</td>
<td>1.00</td>
<td>61.80</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>425 ATOM 63 CG2 ILE A 10</td>
<td>6.343</td>
<td>-4.127</td>
<td>16.180</td>
<td>1.00</td>
<td>61.80</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>426 ATOM 64 CD1 ILE A 10</td>
<td>7.241</td>
<td>-1.341</td>
<td>16.788</td>
<td>1.00</td>
<td>61.80</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>427 ATOM 65 N ASP A 11</td>
<td>3.225</td>
<td>-5.621</td>
<td>15.824</td>
<td>1.00</td>
<td>68.81</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>428 ATOM 66 CA ASP A 11</td>
<td>2.802</td>
<td>-6.960</td>
<td>16.220</td>
<td>1.00</td>
<td>68.81</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>429 ATOM 67 C ASP A 11</td>
<td>3.892</td>
<td>-7.970</td>
<td>15.878</td>
<td>1.00</td>
<td>68.81</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>430 ATOM 68 O ASP A 11</td>
<td>4.569</td>
<td>-7.850</td>
<td>14.850</td>
<td>1.00</td>
<td>68.81</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Structure Files: Silent File

- Rosetta-specific structure file-type
- Useful when storing many structures
  - Easy conversion between PDB <-> Silent files

Snippet from 1qys.out
PDB files

- International standard
- Useful for a small number of structures
- One line per atom
- Readable by PyMol, MOE, Chimera, etc
- Must be processed with clean_pdb.py prior to use in Rosetta

- Full info:  
  www.wwpdb.org/documentation/file-format

Silent files

- Specific to Rosetta
- Useful for archiving many structures
- One line per residue
- Binary files: more compact, but not human-readable
- Silent --> PDB using 
  $ROSETTA/main/source/bin/extract_pdb
  $ROSETTA/main/source/bin/extract_pdb

- Full info:  
  https://www.rosettacommons.org/docs/latest/rosetta Basics/file_types/silent-file
Sequence Files:

- fasta: string of 1-letter amino acid names
- clustal: common for multiple sequence alignments
- grishin: RosettaCM-specific

1qys.fasta

```plaintext
>1QYS:A|PDBID|CHAIN|SEQUENCE
MGDIQVQVNIDDNGKNFDYTYTVTTESELQKVLNELMDYIKKQGAKRVRISITARTKKEAEKFAAILIKVFAELGYNDINVTFDGDTVTVEGQLEGGSLEHHHHHH
```
Application-Specific Files:
• More details later during tutorials

• Res Files: Indicates which residue positions should be designed
• Params File: Custom parameters for small molecules or non-canonical amino acids
• Span File: Defines which residues are in the membrane
• Loops File: Identifies the loop residues for loop closure
• Constraint File: Experimentally derived restraints
• Fragment File: Short protein segments used for comparative modeling and de novo folding
Now we have our input files...what do we do now?

$ROSETTA/main/source/bin/<your favorite Rosetta application> -option1 <argument1> -option2 -option3 <argument3a> <argument3b>...

Note: Some options can take one or multiple arguments, while some don’t take any.
Now we have our input files...what do we do now?

$ROSETTA/main/source/bin/<your favorite Rosetta application> -option1 <argument1> -option2 -option3 <argument3a> <argument3b>...

Note: Some options can take one or multiple arguments, while some don’t take any.

>> $ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease -in:file:fasta ./input_files/1qys_A.fasta -in:file:native ./input_files/1qys.pdb -in:file:frag3 ./input_files/aa1elwA03_05.200_v1_3 -in:file:frag9 ./input_files/aa1elwA09_05.200_v1_3 -abinitio:relax -relax:fast -abinitio::increase_cycles 10 -abinitio::rg_reweight 0.5 -abinitio::rsd_wt_helix 0.5 -abinitio::rsd_wt_loop 0.5 -use_filters true -psipred_ss2 ./input_files/1elwA.psipred_ss2 -kill_hairpins -out:file:silent ./output_files/1qys_10.out -out:file:scorefile ./output_files/1qys_silent_scores.sc -nstruct 10

Above command-line taken directly from: Barth, P., Schonbrun, J., Baker, D. Towards high-resolution prediction and design of transmembrane helical protein structures (2007).
Now we have our input files...what do we do now?

$ROSETTA/main/source/bin/<your favorite Rosetta application> -option1 <argument1> -option2 -option3 <argument3a> <argument3b>...

Note: Some options can take one or multiple arguments, while some don’t take any.

>> $ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease -in:file:fasta ./input_files/1qys_A.fasta -in:file:native ./input_files/1qys.pdb -in:file:frag3 ./input_files/aalelwA03_05.200_v1_3 -in:file:frag9 ./input_files/aalelwA09_05.200_v1_3 -abinitio:relax -relax:fast -abinitio::increase_cycles 10 -abinitio::rg_reweight 0.5 -abinitio::rsd_wt_helix 0.5 -abinitio::rsd_wt_loop 0.5 -use_filters true -psipred_ss2 ./input_files/1elwA.psipred_ss2 -kill_hairpins -out:file:silent ./output_files/1qys_10.out -out:file:scorefile ./output_files/1qys_silent_scores.sc -nstruct 10

Above command-line taken directly from: Barth, P., Schonbrun, J., Baker, D. Towards high-resolution prediction and design of transmembrane helical protein structures (2007).
- Contains all the same options as the command line, just in a more user-friendly fashion
- Easier to read, edit, log, etc.

<table>
<thead>
<tr>
<th>options.txt</th>
</tr>
</thead>
<tbody>
<tr>
<td>## This is a comment</td>
</tr>
<tr>
<td>## input files</td>
</tr>
<tr>
<td>-in:file</td>
</tr>
<tr>
<td>-fasta ./input_files/1qys_A.fasta ## specifies location of fasta sequence file</td>
</tr>
<tr>
<td>-native ./input_files/1qys.pdb ## specifies location of native PDB; &quot;native&quot; files are often used to compare final output structures</td>
</tr>
<tr>
<td>-frag9 ./input_files/aa1elwA03_05.200_v1_9 ## not in this directory, but are 9-mer residue fragments generated on Robetta server</td>
</tr>
<tr>
<td>-frag3 ./input_files/aa1elwA03_05.200_v1_3 ## not in this directory, but are 3-mer residue fragments generated on Robetta server</td>
</tr>
<tr>
<td>-psipred_ss2 ./input_files/1elwA.psipred_ss2 ## not in this directory, but secondary structure prediction outputs from Robetta server</td>
</tr>
<tr>
<td>## application-specific flags</td>
</tr>
<tr>
<td>-abinitio:relax</td>
</tr>
<tr>
<td>-relax:fast</td>
</tr>
<tr>
<td>-abinitio::increase_cycles 10</td>
</tr>
<tr>
<td>-abinitio::rg_reweight 0.5</td>
</tr>
<tr>
<td>-abinitio::rsd_wt_helix 0.5</td>
</tr>
<tr>
<td>-abinitio::rsd_wt_loop 0.5</td>
</tr>
<tr>
<td>-use_filters true</td>
</tr>
<tr>
<td>-kill_hairpins</td>
</tr>
<tr>
<td>## output options</td>
</tr>
<tr>
<td>-out:file:silent ./output_files/1qys_10.out</td>
</tr>
<tr>
<td>-out:file:scorefile ./output_files/1qys_silent_scores.sc</td>
</tr>
<tr>
<td>-nstruct 10</td>
</tr>
</tbody>
</table>

$ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease @options.txt
Basic Pipeline for Running a Rosetta Application

<table>
<thead>
<tr>
<th>Input Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
</tr>
<tr>
<td>Sequence</td>
</tr>
<tr>
<td>Application-Specific</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structures</td>
</tr>
<tr>
<td>Scores</td>
</tr>
<tr>
<td>Log/Tracer</td>
</tr>
</tbody>
</table>

$ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease @options.txt
Score Files:
- tab-delimited file containing all score terms for each output

<table>
<thead>
<tr>
<th>SCORE:</th>
<th>score</th>
<th>fa_atr</th>
<th>fa_rep</th>
<th>fa_sol</th>
<th>fa_intra_rep</th>
<th>fa_elec</th>
<th>ref</th>
<th>time</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-145.658</td>
<td>-416.906</td>
<td>48.038</td>
<td>235.048</td>
<td>1.023</td>
<td>-47.764</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0001</td>
</tr>
<tr>
<td>3</td>
<td>-145.660</td>
<td>-420.838</td>
<td>48.025</td>
<td>238.756</td>
<td>1.060</td>
<td>-46.720</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0002</td>
</tr>
<tr>
<td>4</td>
<td>-147.668</td>
<td>-414.585</td>
<td>46.523</td>
<td>235.308</td>
<td>1.033</td>
<td>-48.171</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0003</td>
</tr>
<tr>
<td>5</td>
<td>-147.324</td>
<td>-413.085</td>
<td>46.068</td>
<td>232.161</td>
<td>1.025</td>
<td>-46.810</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0004</td>
</tr>
<tr>
<td>6</td>
<td>-148.283</td>
<td>-423.099</td>
<td>47.967</td>
<td>241.560</td>
<td>1.046</td>
<td>-48.530</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0005</td>
</tr>
<tr>
<td>7</td>
<td>-147.020</td>
<td>-418.127</td>
<td>47.198</td>
<td>237.669</td>
<td>1.018</td>
<td>-48.364</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0006</td>
</tr>
<tr>
<td>8</td>
<td>-148.368</td>
<td>-417.523</td>
<td>46.620</td>
<td>235.293</td>
<td>1.049</td>
<td>-47.662</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0007</td>
</tr>
<tr>
<td>9</td>
<td>-147.644</td>
<td>-414.716</td>
<td>45.941</td>
<td>232.969</td>
<td>1.038</td>
<td>-46.471</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0008</td>
</tr>
<tr>
<td>10</td>
<td>-147.763</td>
<td>-416.130</td>
<td>46.386</td>
<td>235.877</td>
<td>1.023</td>
<td>-47.914</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0009</td>
</tr>
<tr>
<td>11</td>
<td>-146.560</td>
<td>-421.948</td>
<td>49.483</td>
<td>239.471</td>
<td>1.031</td>
<td>-49.106</td>
<td>...</td>
<td>-12.643</td>
<td>1qys_0010</td>
</tr>
</tbody>
</table>
Structure Files:

- PDB or silent file -- same as input files, but note that output PDBs contain per residue score terms after ATOM lines.

/output_files/1qys_0001.pdb
Log Files/Tracer:

- Printed version of what runs on the terminal
- Keeps track of what's occurring during a simulation--VERY important for keeping records
- These files get large:
  - Silence certain tracers:
    - -mute core.chemical.ResidueTypeSet
  - Change verbosity level
    - -out:levels all:Warning core.init:Info

$ROSETTA/main/source/bin/AbinitioRelax.linuxgccrelease @options.txt >> run.log
Any Questions?
With great power comes great responsibility
cd ~/rosetta_workshop/rosetta/

Basic Rosetta Structure

- main/
- tools/
- demos/
- documentation/
cd ~/rosetta_workshop/rosetta/

Basic Rosetta Structure

- main/
- tools/
- demos/
- documentation/
cd ~/rosetta_workshop/rosetta/main/source/

- **Rosetta/main/source/bin/**
  - Most applications you will run are calling programs within the bin directory
    - rosetta_scripts.default.linuxgccrelease
    - score_jd2.default.linuxgccrelease
    - relax.default.linuxgccrelease

- **Rosetta/main/source/scons.py**
  - Used for compiling

- **Rosetta/main/source/src/**
  - This is where all of the code lives
cd ~/rosetta_workshop/rosetta/main/database/
(relative path from source/ == cd ../database/)

- **Contains pre-defined information that Rosetta needs for runs**
  - **Rosetta/main/database/chemical/**
    - Residue information--params files
    - Atom sets
  - **Rosetta/main/database/scoring/**
    - Default weights files
    - Rotamer libraries
cd
~/rosetta_workshop/rosetta/main/tests/

- **Rosetta/main/tests/**
  - This is for developer use--don’t worry about this (fun to read though!)
These scripts are incredibly helpful for smaller, more basic tasks

Used mainly to setup or analyze runs
Rosetta/tools/protein_tools/scripts/

- **clean_pdb.py**
  - Makes a PDB “Rosetta-proof” and used at the beginning of almost any protocol
- **pdb_renumber.py**
- **score_vs_rmsd.py**
  - Setup for score vs. RMSD plots
- **top_n_percent.py**
  - Extracts tags (protein names) for top given percent of models based on score term
• **Rosetta/tools/perl_tools/**
  • addchain.pl
  • getFastaFromCoords.pl
  • getCAcoords.pl

• **Rosetta/tools/hts_tools/**
  Scripts used mainly to setup high throughput screen runs (> 1000 compounds)
  • make_params.py
  • get_descriptor_data.py
  • setup_screening_project.py
Input files/options are used in protocols--good place to start when you’re doing learning a new method

- **Rosetta/demos/tutorials/**
  - these tutorials
- **Rosetta/demos/public/**
  - used to go through sample procedures
- **Rosetta/demos/protocol_capture/**
  - protocols associated with a publication

**DISCLAIMER: May be out of date, always check Wiki/RosettaCommons/forum for latest information!**
Everything here is also on the RosettaCommons site!

- Understanding general Rosetta concepts
  - Where to find FAQs (How long does this run take?)
  - Options list, file types
  - Protocols you can use
- General structural biology FAQs
  - How do I do X?
Other Rosetta Resources:

- https://www.rosettacommons.org/docs/latest/Home
  - Forum
  - Documentation
  - User guides
  - Everything Rosetta-related you could ever dream of!

GO HAVE FUN!