Creating custom protocols with ROSETTA

Presented by: Hope Woods

hope.woods@vanderbilt.edu

Custom Protocols

- Rosetta applications cover most common tasks
- Sometimes you want to do something different
  - Modify an existing protocol
  - Combine two protocols
  - Make an entire novel protocol
- Or if you just want to run multiple protocols from the same application

How to Make Custom Protocols

- C++ - Directly modify the Rosetta source code
- PyRosetta – Python bindings for directly interacting with Rosetta functions (http://www.pyrosetta.org/)
- RosettaScripts – XML based interface for creating protocols

```
<ROSETTASCRIPTS>
        <SCOREFXNS>
        </SCOREFXNS>
        <RESIDUE_SELECTORS>
        </RESIDUE_SELECTORS>
        <FILTERS>
        </FILTERS>
        <SIMPLE_METRICS>
        </SIMPLE_METRICS>
        <TASKOPERATIONS>
        </TASKOPERATIONS>
        <MOVERS>
        </MOVERS>
        <PROTOCOLS>
        </PROTOCOLS>
        <OUTPUT />
</ROSETTASCRIPTS>
```

- XML “eXtensible Markup Language”
- Consists of large level tags and sub-tags
- Everything not in brackets < > is a comment
- Read from the bottom up

Tip: Run rosetta_scripts without options to get template

Breaking down a tag

```
<MOVERS>
    <PackRotamersMover name="repack1" scorefxn="ref2015" />
</MOVERS>
```

- Name of mover used
- Name assigned to specific version (can be referenced elsewhere in XML)
- Custom settings

Most tags have required settings or default values, always check documentation

Movers

```
<MOVERS>
    <PackRotamersMover name="repack1" scorefxn="ref2015" taskoperations="ifcl,rtrp" />
    <InterfaceAnalyzerMover name="iface" scorefxn="ref2015" fixedchains="A,B" />
</MOVERS>
```

- Movers are the basic building blocks of a RosettaScripts protocol
- Most modify the pose
  - Some compute metrics instead

Movers

```
<MOVERS>
    <PackRotamersMover name="repack1" scorefxn="ref2015" taskoperations="ifcl,rtrp" />
    <PackRotamersMover name="repack2" scorefxn="ref2015" taskoperations="ifcl" />
    <PackRotamersMover name="repack3" scorefxn="ref2015_soft" taskoperations="ifcl" />
    <InterfaceAnalyzerMover name="iface" scorefxn="ref2015" fixedchains="A,B" />
</MOVERS>
```

- Movers are the basic building blocks of a RosettaScripts protocol
- Most modify the pose
  - Some compute metrics instead
- A single mover can be used more than once

Protocols

```
<PROTOCOLS>
      <Add mover="Repack1"/>
      <Add mover="Repack2" filter="avg_deg" />
      <Add mover="iface" />
</PROTOCOLS>
```

- Movers are executed in the order specified in PROTOCOLS
- Movers can be combined with filters
- Movers can be used more than once in a protocol

Score Functions

```
<SCOREFXNS>
    <ScoreFunction name="ref2015" weights="ref2015.wts" />
    <ScoreFunction name="ref2015_soft" weights="ref2015_soft.wts">
        <Reweight scoretype="atom_pair_constraint" weight="1.0" />
    </ScoreFunction>
</SCOREFXNS>
```

- Different parts of a protocol can use different score functions
- Standard score functions can be modified

Filters

```
<FILTERS>
        <ScoreType name="score_type_filter" scorefxn="ref2015" score_type="total_score"
Threshold="-500" />
        <AverageDegree name="avg_deg" threshold="8" distance_threshold="10"
        task_operations="rtiv" />
</FILTERS>
```

- Can pass/fail an output structure
  - Stop a run earlier if the output will be bad.
- Also can be used to compute protein metrics, but Simple Metrics is recommended for this

Filters

```
<FILTERS>
        <ScoreType name="score_type_filter" scorefxn="ref2015" score_type="total_score"
Threshold="-500" />
        <AverageDegree name="avg_deg" threshold="8" distance_threshold="10"
        task_operations="rtiv" />
</FILTERS>
…
<MOVERS>
        …
        <IteratedConvergence name="iter_conv" mover="repack" filter="score_type_filter"
delta="0.1" cycles="10" maxcycles="100" />
</MOVERS>
```

- Can pass/fail an output structure
  - Stop a run earlier if the output will be bad.
- Also can be used to compute protein metrics, but Simple Metrics is recommended for this

Simple Metrics

```
<SIMPLE_METRICS>
      <RMSDMetric name="rmsd" rmsd_type="rmsd_protein_bb_heavy" residue_selector="L1"
use_native="1" />
      <TotalEnergyMetric name="total_energy" residue_selector="L1"/>
</SIMPLE_METRICS>
```

- Used to compute protein metrics
- Can be ran at different points during a protocol and output to a score file
- Use in combination with RunSimpleMetrics mover

Simple Metrics

```
<SIMPLE_METRICS>
    <RMSDMetric name="rmsd" rmsd_type="rmsd_protein_bb_heavy" residue_selector="L1"
use_native="1" />
    <TotalEnergyMetric name="total_energy" residue_selector="L1" />
</SIMPLE_METRICS>
…
<MOVERS>
    …
    <RunSimpleMetrics name="run_metrics1" metrics="total_energy" prefix="m1_" />
</MOVERS>
```

- Used to compute protein metrics
- Can be ran at different points during a protocol and output to a score file
- Use in combination with RunSimpleMetrics mover

Simple Metrics

```
<SIMPLE_METRICS>
        <RMSDMetric name="rmsd" rmsd_type="rmsd_protein_bb_heavy" residue_selector="L1"
use_native="1" />
        <TotalEnergyMetric name="total_energy" residue_selector="L1" />
</SIMPLE_METRICS>
…
<MOVERS>
        …
        <RunSimpleMetrics name="run_metrics1" metrics="total_energy" prefix="m1_" />
</MOVERS>
…
<PROTOCOLS>
        …
        <Add mover="Repack1" />
        <Add mover="run_metrics1" />
        …
</PROTOCOLS>
```

Residue Selectors

```
<RESIDUE_SELECTORs>
      <Chain name="chA" chains="A"/>
       <Index name="res1to10" resnum="1-10"/>
</RESIDUE_SELECTORS>
…
<MOVERS>
      …
      <AwesomeScienceMover name="solve" residue_selector="chA"  />
</MOVERS>
```

- Selects a subset of the system for Rosetta to operate on
- There are overlaps with other XML parts (example: a mover may define residues in its own way)

Task Operations

```
<TASKOPERATIONS>
      <ReadResfile name="rrf" filename="resfile" />
      <RestrictToRepacking name="rtrp" />
      <RestrictResidueToRepacking name="restrict_Y100" resnum="100" />
</TASKOPERATIONS>
…
      <PackRotamersMover name="repack1" taskoperations="rtrp" />
```

- Select residues
- Specify how to allow side chain movement
- Specify which residues to allow to design

Output

```
<OUTPUT scorefxn="ref2015" />
```

- Specifies the score function used for the final output model and in the scorefile
- If you use multiple score functions in a protocol or use a non-default score function – make sure to flag this

Running Rosetta Scripts

rosetta_scripts.linuxgccrelease -parser:protocol protocol.xml

The application

Runs whatever procedure is dictated in the XML file

The actual protocol

The file that describes your experimental steps

Not seen: @options file and command line options

Useful Features

- Rewrite old Rosetta XML scripts
  - tools/xsd_xrw/rewrite_rosetta_scripts.py
- Validate your XML scripts
  - https://www.rosettacommons.org/docs/latest/application_documentation/rosetta_scripts/validate_rosetta_script
- Variable substitutions
  - -parser:script_vars repeat=5 cutoff=10.0 on command line changes every %%repeat%% to 5 and %%cutoff%% to 10.0 in XML

Documentation

RosettaScripts documentation

https://www.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/RosettaScripts

Possible Movers**

https://www.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Movers/Movers-RosettaScripts

Original reference

Fleishman, Sarel J., et al. "RosettaScripts: a scripting language interface to the Rosetta macromolecular modeling suite." PloS one 6.6 (2011): e20161.

**Best place to start