

Creating custom protocols with ROSETTA

Alberto Cisneros

April 2014

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

How to Make Custom Protocols

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

ROSETTAScripts

- XML is “eXtensible Markup Language”
- Widely used for representing hierarchical data
- Rosetta supports a subset of XM

Structure of ROSETTA XML file

<ROSETTASCRIPTS>

<SCOREFXNS>

</SCOREFXNS>

<FILTERS>

</FILTERS>

<TASKOPERATIONS>

</TASKOPERATIONS>

<MOVERS>

</MOVERS>

<PROTOCOLS>

</PROTOCOLS>

</ROSETTASCRIPTS>

- SCOREFXN

- Definitions of score functions

- FILTERS

- Definitions of filters

- TASKOPERATIONS

- Definition of Task Operations

- MOVERS

- Definition of movers

- PROTOCOLS

- Set of protocols to execute

Score Functions

```
<SCOREFXNS>  
  <ligand_soft_rep weights="ligand_soft_rep"/>  
  <hard_rep weights="ligand">  
    <Reweight scoretype="fa_intra_rep" weight="0.004"/>  
    <Reweight scoretype="hack_elec" weight="0.42"/>  
  </hard_rep>  
</SCOREFXNS>
```

- Different parts of a protocol can use different score functions
- Standard score functions can be modified
- Filters and movers have access to all defined functions

Task Operations

<TASKOPERATIONS>

<ReadResfile name="rrf" />

<ReadResfile name="rrf2" filename="resfile2" />

<RestrictToRepacking name="rtrp" />

<RestrictResidueToRepacking name="restrict_Y100" resnum="100" />

</TASKOPERATIONS>

- Define how protein side chains are packed
- Select residues
- Calculate protein metrics
- Useful if you only want to repack part of protein
- Task Operations are available to all movers
 - Not all movers obey them

Filters

```
<FILTERS>
```

```
  <ScoreType name="score_type_filter" scorefn="score12" score_type="total_score"  
Threshold="-500" />
```

```
  <AverageDegree name="avg_deg" threshold=8 distance_threshold=10  
task_operations="rtiv" />
```

```
</FILTERS>
```

- These determine whether or not a pose is output
- Filters only look at current pose
- Some can be used to compute protein metrics
- Can potentially over-constrain your protocol

Movers

```
<MOVERS>
```

```
<PackRotamersMover name="repack1" scorefxn="score12_002" taskoperations="ifcl,rtrp" />  
<PackRotamersMover name="repack2" scorefxn="score12_005" taskoperations="ifcl,rtrp" />  
<PackRotamersMover name="repack3" scorefxn="score12_055" taskoperations="ifcl,rtrp" />  
<InterfaceAnalyzerMover name="iface" scorefxn="score12" 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

How ROSETTA Scripts parses protocols

- At initialization
 - Movers, filters, scoring functions, etc are initialized
- For each input job
 - Movers and filters are executed in the order specified in PROTOCOLS
- Movers are not aware of one another
- Jobs are not aware of one another

Executing Protocols

```
rosetta_scripts.default.linuxgccrelease @flags.txt -parser:protocol protocol.xml
```

- RosettaScripts protocols are run using the rosetta_scripts application
- All the normal file IO options are supported
- Some movers have their own associated command line flags

Documentation

- For more information on RosettaScripts, documentation is available here:
 - https://www.rosettacommons.org/manuals/archive/rosetta3.5_user_guide/RosettaScripts_Documentation.html