

Creating custom protocols with ROSETTA

Darwin Fu

darwinyfu@gmail.com

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 (new format since last year)

Running Rosetta Scripts

```
rosetta_scripts.linuxgccrelease -parser:protocol protocol.xml
```



The application

Runs whatever
procedure is dictated
in the XML file

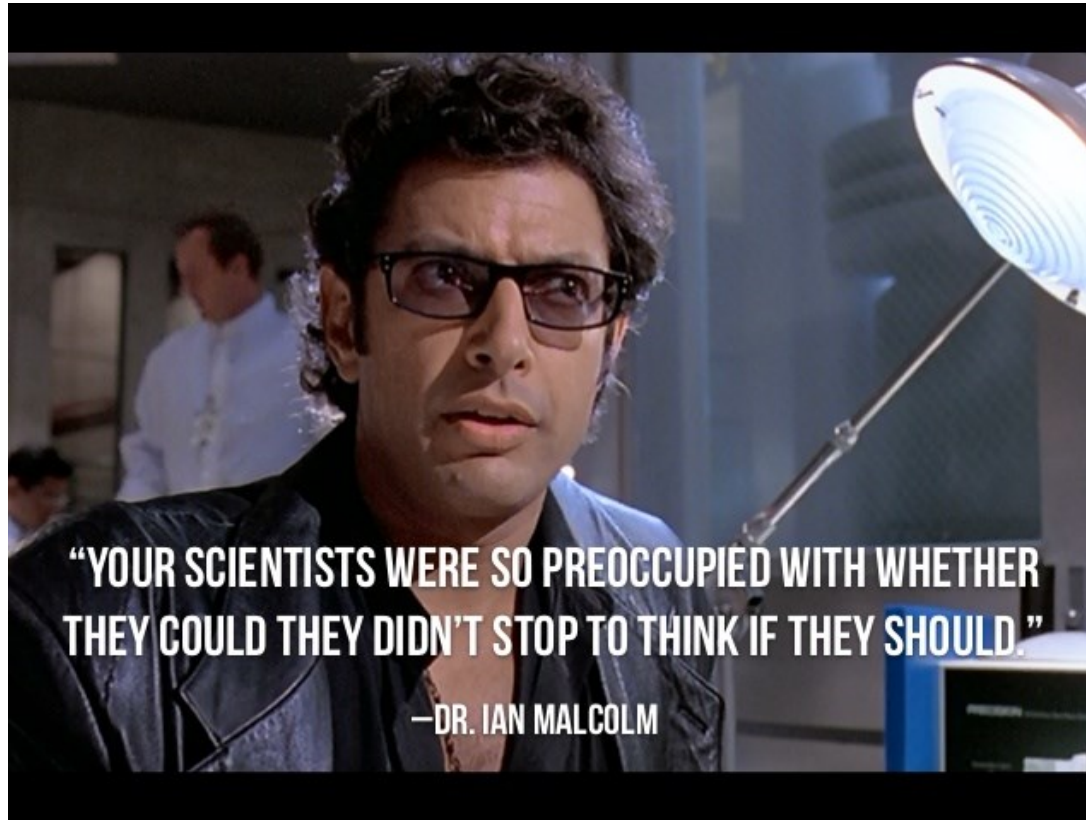


The actual protocol

The file that
describes your
experimental steps

Not seen: @options file and command line options

A Fair Warning



Not all XMLs/options have been tested in combination
Some XMLs require specific options to work

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

- XML “eXtensible Markup Language”
- Consists of large level tags and sub-tags
- Widely used for representing hierarchical data
- Everything not in brackets
< > is a comment

Tip: Run rosetta_scripts
without options to get
template

Breaking down a tag

<MOVERS>

 <PackRotamersMover name="repack1" scorefxn="score12_002" />

</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="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

Filters

```
<FILTERS>
```

```
  <ScoreType name="score_type_filter" scorefxn="score12" 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

Score Functions

```
<SCOREFXNS>
```

```
  <ScoreFunction name="ligand_soft_rep" weights="ligand_soft_rep" />
```

```
  <ScoreFunction name="hard_rep" weights="ligand">
```

```
    <Reweight scoretype="fa_intra_rep" weight="0.004"/>
```

```
    <Reweight scoretype="fa_elec" weight="0.42"/>
```

```
  </ScoreFunction>
```

```
</SCOREFXNS>
```

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

Residue Selectors

```
<RESIDUE_SELECTORS>
```

```
  <Chain name="chA" chains="A"/>
```

```
  <Index name="res1to10" resnum="1-10"/>
```

```
</RESIDUE_SELECTORS>
```

```
<AwesomeScienceMover name="solve" residue_selector="chA" />
```

- 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 sidechain movement
- Specify which residues to allow to design

Apply To Pose

```
<APPLY_TO_POSE>  
  <profile weight="1.0" file_name="constraints.cst" />  
</APPLY_TO_POSE>
```

- Applies a broad change (most likely constraints) to input structure
- Does not need to be listed in the PROTOCOLS section

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

Output

```
<OUTPUT scorefxn="talaris2014" />
```

- 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!

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

The top of this help page contains links to individual listings for Movers/Filters/TaskOperations...etc.

Original reference

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