RosettaScripts XML

Samuel Schmitz (Samuel.Schmitz@vanderbilt.edu)

August 6th, 2016

Overview

- RosettaScripts is a XML interface to the Rosetta libraries.
- Allows development with Rosetta without using C++
- Fast and easy to develop

- ScoreFunctions ...
- TaskOperations ...
- Movers ...
- Filters ...

  … can easily be recombined to create protocols for specialized tasks.

Skeleton XML Format

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

- XML - “e**X**tensible **M**arkup **L**anguage”
- Widely used for representing hierarchical data
- RosettaScripts supports a variant of XML
- Use **-print_template_script** as starting point for custom Rosetta scripts
- Everything outside the <> notation is being ignored and can be used for comments

```
$ rosetta_scripts -print_template_script -mute core | egrep -v “(core|app)” > rscript.xml
```

Example XML File

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <interface weights=interface/>
  </SCOREFXNS>
  <FILTERS>
    <AlaScan name=scan/>
    <Ddg name=ddg/>
    <Sasa name=sasa/>
  </FILTERS>
  <MOVERS>
    <Docking name=dock fullatom=1 local_refine=1 score_high=soft_rep/>
  </MOVERS>
  <APPLY_TO_POSE>
  </APPLY_TO_POSE>
  <PROTOCOLS>
    <Add mover_name=dock filter_name=scan/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
  </PROTOCOLS>
  <OUTPUT scorefxn=interface />
</ROSETTASCRIPTS>
```

- Definition of components you want to use
- Parametrization

Example XML File

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <interface weights=interface/>
  </SCOREFXNS>
  <FILTERS>
    <AlaScan name=scan/>
    <Ddg name=ddg/>
    <Sasa name=sasa/>
  </FILTERS>
  <MOVERS>
    <Docking name=dock fullatom=1 local_refine=1 score_high=soft_rep/>
  </MOVERS>
  <APPLY_TO_POSE>
  </APPLY_TO_POSE>
  <PROTOCOLS>
    <Add mover_name=dock filter_name=scan/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
  </PROTOCOLS>
  <OUTPUT scorefxn=interface />
</ROSETTASCRIPTS>
```

- Definition of components you want to use
- Parametrization

- To change the input before the protocol starts

Example XML File

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <interface weights=interface/>
  </SCOREFXNS>
  <FILTERS>
    <AlaScan name=scan/>
    <Ddg name=ddg/>
    <Sasa name=sasa/>
  </FILTERS>
  <MOVERS>
    <Docking name=dock fullatom=1 local_refine=1 score_high=soft_rep/>
  </MOVERS>
  <APPLY_TO_POSE>
  </APPLY_TO_POSE>
  <PROTOCOLS>
    <Add mover_name=dock filter_name=scan/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
  </PROTOCOLS>
  <OUTPUT scorefxn=interface />
</ROSETTASCRIPTS>
```

- Definition of components you want to use
- Parametrization

- To change the input before the protocol starts

Example XML File

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <interface weights=interface/>
  </SCOREFXNS>
  <FILTERS>
    <AlaScan name=scan/>
    <Ddg name=ddg/>
    <Sasa name=sasa/>
  </FILTERS>
  <MOVERS>
    <Docking name=dock fullatom=1 local_refine=1 score_high=soft_rep/>
  </MOVERS>
  <APPLY_TO_POSE>
  </APPLY_TO_POSE>
  <PROTOCOLS>
    <Add mover_name=dock filter_name=scan/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
  </PROTOCOLS>
  <OUTPUT scorefxn=interface />
</ROSETTASCRIPTS>
```

- Definition of components you want to use
- Parametrization

- To change the input before the protocol starts

- Score the structure before writing it to disk

Example XML File

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <interface weights=interface/>
  </SCOREFXNS>
  <FILTERS>
    <AlaScan name=scan/>
    <Ddg name=ddg/>
    <Sasa name=sasa/>
  </FILTERS>
  <MOVERS>
    <Docking name=dock fullatom=1 local_refine=1 score_high=soft_rep/>
  </MOVERS>
  <APPLY_TO_POSE>
  </APPLY_TO_POSE>
  <PROTOCOLS>
    <Add mover_name=dock filter_name=scan/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
  </PROTOCOLS>
  <OUTPUT scorefxn=interface />
</ROSETTASCRIPTS>
```

- Definition of components you want to use
- Parametrization

- To change the input before the protocol starts

- Score the structure before writing it to disk

Example Commandline

- rosetta_scripts is the C++ interpreter used for parsing RosettaScripts XML files
- Invalid XML files and/or missing required options will lead to a crash
- rosetta_scripts uses JobDistributor2 for parallelization. JD2 options are compatible with RosettaScripts
- Script can be modified by command line arguments
  `-parser:script_vars cutoff=10.0` will replace all `%%cutoff%%` with `10.0`

```
$ Rosetta/main/source/bin/rosetta_scripts.linuxgccrelease \
    -s <INPUT_PDB_FILE_NAME> \
    -use_input_sc \
    -nstruct 20 \
    -jd2:ntrials 2 \
    -database Rosetta/main/database/ \
    -ex1 -ex2 \
    -parser:view \
    -parser:protocol ala_scan.xml
```

Example Output

- Mute/Unmute Tracer channels to watch ParsedProtocol doing the work

```
$ rosetta_scripts -mute all -unmute protocols.rosetta_scripts -protocols:parser file.xml
```

```
protocols.rosetta_scripts.RosettaScriptsParser: dock_design_filename=ala_scan.xml
protocols.rosetta_scripts.RosettaScriptsParser: Parsed script:
<ROSETTASCRIPTS> … </ROSETTASCRIPTS>
protocols.rosetta_scripts.RosettaScriptsParser: Defined filter named "scan" of type AlaScan
protocols.rosetta_scripts.RosettaScriptsParser: Defined filter named "ddg" of type Ddg
protocols.rosetta_scripts.RosettaScriptsParser: Defined filter named "sasa" of type Sasa
protocols.rosetta_scripts.RosettaScriptsParser: Defined mover named "dock" of type Docking
protocols.rosetta_scripts.ParsedProtocol: ParsedProtocol mover with the following movers and
filters
protocols.rosetta_scripts.ParsedProtocol: added mover "dock" with filter "scan"
protocols.rosetta_scripts.ParsedProtocol: added mover "NULL_MOVER" with filter "ddg"
protocols.rosetta_scripts.ParsedProtocol: added mover "NULL_MOVER" with filter "sasa"
protocols.rosetta_scripts.ParsedProtocol: ==BEGIN MOVER Docking - dock ==
protocols.rosetta_scripts.ParsedProtocol: ==BEGIN FILTER scan ==
protocols.rosetta_scripts.ParsedProtocol: ==END FILTER scan ==
protocols.rosetta_scripts.ParsedProtocol: ==BEGIN MOVER NullMover - NULL_MOVER ==
protocols.rosetta_scripts.ParsedProtocol: == BEGIN FILTER ddg ===
protocols.rosetta_scripts.ParsedProtocol: == END FILTER ddg ==
protocols.rosetta_scripts.ParsedProtocol: ==BEGIN MOVER NullMover - NULL_MOVER ==
protocols.rosetta_scripts.ParsedProtocol: == BEGIN FILTER sasa ==
protocols.rosetta_scripts.ParsedProtocol: ==END FILTER sasa ==
protocols.rosetta_scripts.ParsedProtocol: setting status to success
```

Example Output

```
protocols.rosetta_scripts.ParsedProtocol.REPORT: == Begin report for scan ==
protocols.rosetta_scripts.ParsedProtocol.REPORT:  LYS 24 A :    0.3126
 SER 25 A :   -0.0781
 ALA 27 A :   -0.0661
...
protocols.rosetta_scripts.ParsedProtocol.REPORT: == End report for scan ==
protocols.rosetta_scripts.ParsedProtocol.REPORT: == Begin report for ddg ==
protocols.rosetta_scripts.ParsedProtocol.REPORT: ddg -31.7952
Compound filter returns: 1
============End report for ddg==================
protocols.rosetta_scripts.ParsedProtocol.REPORT: == Begin report for sasa ==
protocols.rosetta_scripts.ParsedProtocol.REPORT: Sasa= 1538.95
Compound filter returns: 1
============End report for sasa==================
```

Additional output:

- score.sc (Poses scored with scorefxn defined in OUTPUT section)
- PDB_NAME_0001.pdb (<Input file name>_<nstruct number>.pdb)
- ...

Skeleton XML Format

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

- XML - “e**X**tensible **M**arkup **L**anguage”
- Widely used for representing hierarchical data
- RosettaScripts supports a variant of XML
- Use **-print_template_script** as starting point for custom Rosetta scripts

```
$ rosetta_scripts -print_template_script -mute core | egrep -v “(core|app)” > rscript.xml
```

Example XML File

```
<ROSETTASCRIPTS>
  <SCOREFXNS>
    <interface weights=interface/>
  </SCOREFXNS>
  <FILTERS>
    <AlaScan name=scan/>
    <Ddg name=ddg/>
    <Sasa name=sasa/>
  </FILTERS>
  <MOVERS>
    <Docking name=dock fullatom=1 local_refine=1 score_high=soft_rep/>
  </MOVERS>
  <APPLY_TO_POSE>
  </APPLY_TO_POSE>
  <PROTOCOLS>
    <Add mover_name=dock filter_name=scan/>
    <Add filter_name=ddg/>
    <Add filter_name=sasa/>
  </PROTOCOLS>
  <OUTPUT scorefxn=interface />
</ROSETTASCRIPTS>
```

- Definition of components you want to use
- Parametrization

- To change the input before the protocol starts

- Score the structure before writing it to disk

Section SCOREFXNS

- Six predefined set of scoring functions comes with Rosetta
- Add custom scoring functions using SCOREFXNS section

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

- Different score functions can be used for different parts of the protocol

Section TASKOPERATIONS

- Selects residues
- Specifies how to allow sidechain movement
- Specifies which residues are allowed to design

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

Section RESIDUESELECTORS

- Defines a subset of Residues of a Pose
- Residue Level TaskOperation: Combine a ResidueSelector with a Residue Level TaskOperation

```
<RESIDUE_SELECTORS>
        <Chain name=chA chains=A/>
        <Index name=1to10 resnums=1-10/>
</RESIDUE_SELECTORS>
```

Residue Level TASKOPERATIONS

- **OperateOnResidueSubset**: Combine a ResidueSelector with a Residue Level TaskOperation

```
<TASKOPERATIONS>
    <OperateOnResidueSubset name=”RepackChainA” selector=chA/>
        <RestricktToRepackingRLT/>
    </OperateOnResidueSubset>
</TASKOPERATIONS>
```

Section FILTERS

- Accepts/Rejects Poses
- Aborts the protocol upon rejection

```
<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>
```

Section MOVERS

- Basic building block of RosettaScripts
- Usually modify the Pose
- Single can be used multiple times

```
<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>
```

Section PROTOCOLS

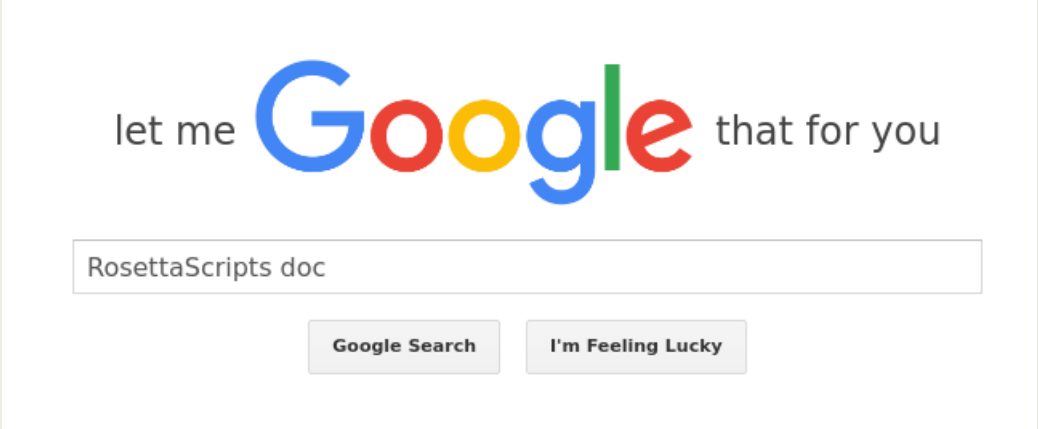
- PROTOCOLS section carries out the order of operations
- Filters can be attached to Movers
- Movers can be used more than once

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

Documentation

- Documentation available for RosettaScripts, Movers, Filters ... including available options at

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



Documentation

- Documentation available for RosettaScripts, Movers, Filters ... including available options at

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

- 41 Filters (Score, Distance, Burial, Bonding …)
- 150+ Movers
- Null-Mover
- Meta-Movers
- General Movers (Constraints, Backbone, KIC)
- Antibody Modeling
- Carbohydrate Specific
- Affinity Maturation mover
- DNA interface design
- Ligand centric
- Loop Modeling
- Protein Interface
- ...