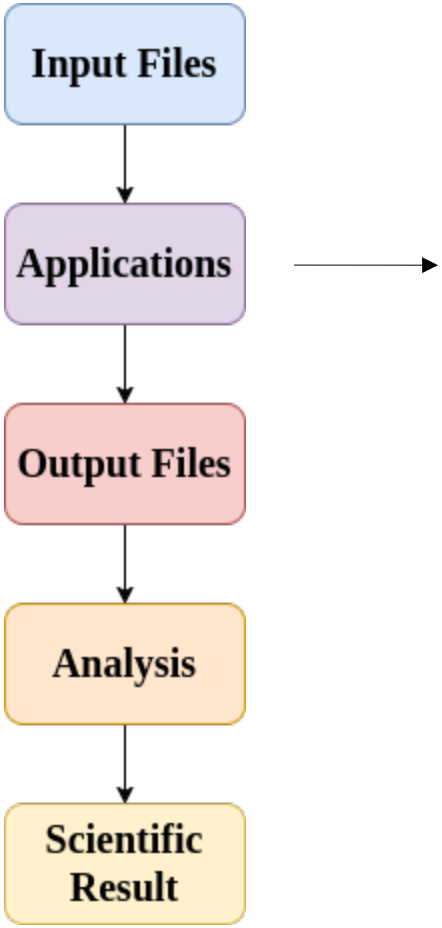
Rosetta XML

Cristina Elisa Martina
2025 Rosetta Workshop
Meiler Lab

Applications



Applications (old school)

.../main/source/bin/**<app_name>**.default.linuxgccrelease

AbinitioRelax
docking_protocol
Fixbb
antibody_designer

Movers/XML files (default)

.../main/source/bin/**rosetta_scripts**.default.linuxgccrelease \
-parser:protocol XXX.xml

DockingProtocolMover
FixBBMover
FastDesignMover

Why moving away from applications?

- Difficult cross-talk between applications
(i.e. different options for each application)

- Limited flexibility to:
Modify existing protocols (i.e. custom protocols)
Combine 2 or more protocol in one single run
(i.e. design + docking)
Create novel protocols for your specific system

How to make Custom Protocols?

- C++: Directly modify the Rosetta source code
(good luck with that!)

- PyRosetta: Python bindings to directly interact with Rosetta functions (http://www.pyrosetta.org/). Annual workshops organized by the Rosetta community!

- RosettaScripts (XMLs): XML based interface for creating protocols. Do not require major scripting skills (aka, easy to use for wet-lab folks)

How to run XML scripts

.../rosetta_scripts.default.linuxgccrelease -parser:protocol XXX.xml

The application:

It will run whatever protocol is reported in the XML file

The actual protocol:

It contains the instructions for your experimental steps

Not shown, but your command line for XML will contain also @options.txt!

A fair warning

New protocols have been tested and showed evidence that they fulfill their task (benchmarking)

However:
- Not all XMLs/options have been tested in combination
- Some XMLs require specific options to work (check documentation)

XML scripts

```
<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

- Widely used to represent hirarchical data

- Consist in levels of tags and subtags

- Everything outside brackets **< >** is a comment

- You can get the empty template running “rosetta_scripts” without options!

XML scripts – easy case (long)

```
<ROSETTASCRIPTS>
            <SCOREFXNS>
            </SCOREFXNS>
            <RESIDUE_SELECTORS>
                        <Index name="align" resnums="1-152"/>
                        <Index name="align_native" resnums="1-152"/>
            </RESIDUE_SELECTORS>
            <SIMPLE_METRICS>
                        <RMSDMetric name="RMSD" residue_selector="align" use_native="1" \
                        residue_selector_ref="align_native" robust="true" super="1"     \
                        rmsd_type="rmsd_all" />
            </SIMPLE_METRICS>
            <TASKOPERATIONS>
            </TASKOPERATIONS>
            <FILTERS>
            </FILTERS>
            <MOVERS>
                        <RunSimpleMetrics name="run_metrics1" metrics="RMSD" prefix="m1_" />
            </MOVERS>
            <APPLY_TO_POSE>
            </APPLY_TO_POSE>
            <PROTOCOLS>
                        <Add mover="run_metrics1"/>
            </PROTOCOLS>
            <OUTPUT />
</ROSETTASCRIPTS>
```

XML scripts – easy case (short)

```
<ROSETTASCRIPTS>
            <RESIDUE_SELECTORS>
                        <Index name="align" resnums="1-152"/>
                        <Index name="align_native" resnums="1-152"/>
            </RESIDUE_SELECTORS>
            <SIMPLE_METRICS>
                        <RMSDMetric name="RMSD" residue_selector="align" use_native="1" \
                        residue_selector_ref="align_native" robust="true" super="1"     \
                        rmsd_type="rmsd_all" />
            </SIMPLE_METRICS>
            <MOVERS>
                        <RunSimpleMetrics name="run_metrics1" metrics="RMSD" prefix="m1_" />
            </MOVERS>
            <PROTOCOLS>
                        <Add mover="run_metrics1"/>
            </PROTOCOLS>
</ROSETTASCRIPTS>
```

Breaking down tags

```
<ROSETTASCRIPTS>
            <RESIDUE_SELECTORS>
                        <Index name="align" resnums="1-152"/>
                        <Index name="align_native" resnums="1-152"/>
            </RESIDUE_SELECTORS>
            <SIMPLE_METRICS>
                        <RMSDMetric name="RMSD" residue_selector="align" use_native="1" \
                        residue_selector_ref="align_native" robust="true" super="1"     \
                        rmsd_type="rmsd_all" />
            </SIMPLE_METRICS>
            <MOVERS>
                        <RunSimpleMetrics name="run_metrics1" metrics="RMSD" prefix="m1_" />
            </MOVERS>
            <PROTOCOLS>
                        <Add mover="run_metrics1"/>
            </PROTOCOLS>
</ROSETTASCRIPTS>
```

Name of task/mover Name of options Custom setting

Movers

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

AI summary:
A "mover" is a functional unit that **modifies a protein structure** (represented as a "pose") by **performing specific operations** like changing sidechain rotations, backbone conformations, or applying design constraints.

- Basic building block of a protocol
- Modify the pose (majority)
- Compute metrics (minority)
- A single mover can be used more than once

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

Some movers have few options:

```
<ScoreMover name="(&string;)" scorefxn="(&string;)" verbose="(&bool;)" />
```

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Movers/movers_pages/analysis/ScoreMover

Some movers have tons of options:

```
<AntibodyDesignMover name="(&string;)" min_scorefxn="(&string;)"
        atom_pair_cst_weight="(&real;)" dihedral_cst_weight="(&real;)"
        global_atom_pair_cst_scoring="(&bool;)" scorefxn="(&string;)"
        global_dihedral_cst_scoring="(&bool;)" light_chain="(&string;)"
        seq_design_cdrs="(&string;)" graft_design_cdrs="(&string;)"
        mintype="(&string;)" instruction_file="(&string;)"
        mc_optimize_dG="(false &bool;)" mc_interface_weight="(1.0 &real;)"
        mc_total_weight="(0.0 &real;)" do_dock="(&bool;)"
        dock_first_cycles="(2 &positive_integer;)"
        dock_second_cycles="(2 &positive_integer;)" use_epitope_csts="(&bool;
        epitope_residues="(&string;)" paratope_cdrs="(&string;)"
        random_start="(&bool;)"
        design_protocol="(EVEN_CLUSTER_MC &ABdesign_protocols;)"
        primary_cdrs="(&string;)" dock_cycles="(&non_negative_integer;)"
        interface_dis="(&real;)" neighbor_dis="(&real;)"
        outer_cycles="(&non_negative_integer;)"
        relax_cycles="(&non_negative_integer;)"
        inner_cycles="(&non_negative_integer;)"
        mutate_framework_for_cluster="(&bool;)" outer_kt="(&real;)"
        inner_kt="(&real;)" top_designs="(&non_negative_integer;)"
        run_AIM="(&bool;)" remove_antigen="(&bool;)" />
```

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Movers/movers_pages/antibodies/AntibodyDesignMover

Filters

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

AI summary:
A "Rosetta filter" refers to a specialized filter acting as a **quality check** mechanism to assess and **discard protein structures** that don't meet specific criteria during the simulation process, allowing only the most promising candidates to proceed further.

- Can pass/fail an output structure

- Can be used to compute protein metrics

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Filters/Filters-RosettaScripts

Filters examples

ResidueDistance

What is the distance between two residues? Based on each residue's neighbor atom (usually Cbeta)

```
<ResidueDistance name="(&string)" res1_res_num="(&string)" res1_pdb_num="(&st
```

Either *res_num or *pdb_num may be specified for res1 and res2. See RosettaScripts#rosettascripts-conventions_specifying-residues .

ExposedHydrophobics

Computes the SASA for each hydrophobic residue (A, F, I, M, L, W, V, Y). The score returned reflects both the number of solvent-exposed hydrophobic residues and the degree to which they are exposed. The score is calculated as follows. For each hydrophobic residue, if the SASA is above a certain cutoff value (default=20), then the value of (SASA - sasa_cutoff) is added to the calculated score. The filter passes if the calculated score is less than the user-specified threshold.

```
<ExposedHydrophobics name="(&string)" sasa_cutoff="(20 &Real)" threshold="(-1
```

- sasa_cutoff: If a residue has SASA lower than this value, it is considered buried and does not affect the score returned by the ExposedHydrophobics filter.
- threshold: If a protein has an ExposedHydrophobics total score below this value, it passes the filter. If a negative threshold is specified, the filter will always pass.

ResidueCount

Filters structures based on the total number of residues in the structure.

```
<ResidueCount name="(&string)" max_residue_count="(Inf &Integer)" min_residue
```

- residue_types: Comma-separated list of which residue type names. (e.g. "CYS,SER,HIS_D"). Only residues with type names matching those in the list will be counted.
- include_property: Comma-separated list of properties (e.g. "HYDROPHOBIC,ALIPHATIC"). Residues with any of these properties will be counted.
- max_residue_count: Is the total number of residues less than or equal to the maximum allowable residue count?
- min_residue_count: Is the total number of residues more than or equal to the minimum allowable residue count?
- count_as_percentage: If this is true, count residues as percentage (=100*raw_number_of_specified_residue/total_residue) instead of counting raw number of it, also max_residue_count/min_residue_count are assumed to be entered as percentage
- residue_selector: Name of a residue selector which defines a subset of residues over which to perform calculation. Default is all residues.
- task_operations: defines subset of residues over which to perform calculation, default only check designable residues
- packable: T/F, also count repackable residues in task_operations

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/Filters/Filters-RosettaScripts

Score functions

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

AI summary:
A Rosetta score function is a computational method used to **assign a numerical "score"** to a protein structure, representing its **predicted stability** based on a weighted sum of various **energy terms** like van der Waals interactions, hydrogen bonds, and backbone geometry, essentially acting as a way to rank different protein conformations based on their predicted likelihood of occurring in nature; **the higher the score, the more energetically favorable the structure is considered to be.**

Wrong! Don't trust AI, the lower the score the more favorable!!!

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/RosettaScripts#rosettascript-sections_scorefunctions

Score functions

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

- Different parts of a protocol can use different score functions (you will see it in the protein docking tutorial!)

- Standard score function can be modified

- Here you can change weights for different energy terms if needed

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

Residue Selectors

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

AI summary:
A "residue selector" is a tool that allows users to **specify a subset of residues** from a protein structure to be specifically targeted or manipulated during a simulation, essentially **defining** which parts of the protein to **focus**.

- Define subsets of the system for Rosetta to operate

- There are overlaps with other XML parts that can also request to define sub-sets of residues

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

Residue Selectors examples

NotResidueSelector

```
<Not name="(&string)" selector="(&string)">
```

CDRResidueSelector

Selects CDR residues in an antibody or camelid antibody.

- Author: Dr. Jared Adolf-Bryfogle (jadolfbr@gmail.com)
- PIs: Dr. Roland Dunbrack and Dr. William Schief

```
<CDR name="(&string)" cdrs="(&string,&string)" input_ab_scheme="(&string)"
```

- *cdrs (&string,&string)* (default=all cdrs): Select the set of CDRs you wish to restrict to (ex: H1 or h1)
- *input_ab_scheme (&string)*: Set the antibody numbering scheme. Must also set the cdr_definition XML option. Both options can also be set through the command line (recommended). See General Antibody Tips for more info.
- *cdr_definition (&string)*: Set the cdr definition you want to use. Must also set the numbering_scheme XML option.

SymmetricalResidueSelector

```
<SymmetricalResidue name=(%string) selector=(%string) />
```

The SymmetricalResidueSelector, when given a selector, will return all symmetrical copies (including the original selection) of those residues. While the packer is symmetry aware, not all filters are. This selector is useful when you need to explicitly give residue numbers but you are not sure which symmetry subunit you need.

ResidueIndexSelector

```
<Index name="(&string)" resnums="(&string)" error_on_out_of_bounds_index="(tr
```

- The string given for the "resnums" option should be a comma-separated list of residue identifiers
- Each residue identifier should be either:
  - *an integer* , so that the Pose numbering can be used,
  - *two integers separated by a dash* , designating a range of Pose-numbered residues,
  - *an integer followed by a single character* , e.g. 12A, referring to the PDB numbering for residue 12 on chain A,
  - *an integer followed by a single character, followed by a dash, followed by an integer followed by a single character*, e.g. 12A-47A, referring to residues 12 through 47 on chain A in PDB numbering. (Note, residues that contain insertion codes cannot be properly identified by these PDB numbered schemes).
- The ResidueIndexSelector sets the positions corresponding to the residues given in the resnums string to true, and all other positions to false.
- If "error_on_out_of_bounds_index" is true (the default), this selector throws an error if a user attempts to select a residue index that doesn't exist in the pose (*e.g.* residue 56 of a 55-residue pose). This behaviour can be disabled by setting "error_on_out_of_bounds_index" to false, which allows the selector to ignore indices that are out of range silently.
- If reverse is true(default false) the residue 1 index is for the last protein residue(soon to be checked in)

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

Task Operations

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

AI summary:
A "task operation" refers to a **specific instruction** that defines **how** a particular set of residues should be **manipulated** or **designed** during a protein modeling simulation, typically used to optimize sidechain conformations; essentially, it allows for **targeted modifications** to specific parts of a protein structure based on user-defined criteria.

- Select residues
- Specify sidechains movement
- Specify designable residues

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/TaskOperations/TaskOperations-RosettaScripts

Task Operations

Per Residue Specification

TaskOp	Description
OperateOnResidueSubset	Use ResidueSelectors and Residue Level TaskOperations to specify a particular behavior on a particular subset of residues.
OperateOnCertainResidues Operation	An older way of specifying particular groups of residues. **Deprecated** - use OperateOnResidueSubset instead.

Packer Behavior Modification

TaskOp	Description
KeepSequenceSymmetry	Prevent chains from differing in sequence
ModifyAnnealer	Change the behavior of the packer.
ProteinLigandInterfaceUpweighter	Increase the contribution of protein/ligand interactions during design.
RestrictInteractionGraphThreadsOperation	Limit the number of threads that the packer may use for interaction graph pre-calculation. (Only affects behaviour of multi-threaded build.)
SetIGType	Set the type of interaction graph to use

Property-based specification

TaskOp	Description
RestrictToResidueProperties	Restrict the palette of `ResidueType` s to those with the given properties.
ProhibitResidueProperties	Prohibit `ResidueType` s with the given properties from the palette.
ConservativeDesign	Only design to amino acids that are similar to native.
ConsensusLoopDesign	Only design to amino acids in loops which match the ABEGO torsion bins.
DsspDesign	Specify design identity based on secondary structure.
DesignCatalyticResidues	Only design residues surrounding the residues in enzdes constraints.
DesignByResidueCentrality	Design only residues which have a high inter-connectedness to other residues.
DesignByCavityProximity	Only design residues around voids.
DesignRandomRegion	Design only a random section of the pose.
HighestEnergyRegion	Design only residues which have a bad energy.
DesignRestrictions	Set allowable AAs and/or residue_level_operations for multiple residue selectors. Use instead of LayerDesign.
LayerDesign	LEGACY: Specify design identity based on secondary structure and burial.
NoRepackDisulfides	Do not repack disulfide residues.
ProteinCore	Do not design residues in the protein core.
SelectResiduesWithinChain	Do not pack/design residues based on their position in a chain.
SelectBySASA	Repack residue based on surface exposure.

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/TaskOperations/TaskOperations-RosettaScripts

Protocols

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

AI summary:
A "protocol" refers to a specific **set of instructions** designed to **perform a** particular molecular modeling **task** on a protein structure, such as protein structure prediction, protein design, protein-protein docking, or ligand binding site identification, by **defining** a series of steps with specific **parameters** to guide the simulation and generate different structural models. Essentially, **it's a recipe for how to use Rosetta** to tackle a specific biological question by manipulating protein structures through various computational methods.

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

Protocols

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

- Main section of a XML
- It defines each steps to be performed
- Includes both movers and filters
- Same mover can be called multiple times
- Movers are NOT aware of one another
- Jobs are NOT aware of one another
- Only YOU are aware
- The order matters!

Protocols example

```
<ROSETTASCRIPTS>
    ...
    <MOVERS>
        <PackRotamersMover name="repack" scorefxn="REF2015" task_operations="ifcl,rtr" />
        <MinMover name="minimize_sc" scorefxn="REF2015" chi="1" bb="0" tolerance="0.001"     \
            jump="0" type="dfpmin_armijo_nonmonotone" />
        <MinMover name="Minim" jump="ALL" cartesian="0" bondangle="0" omega="0"              \
            bondlength="0" chi="1" bb="1" />
        <FastRelax name="FR" scorefxn="REF2015" disable_design="1" task_operations="ifcl"/>
        <InterfaceAnalyzerMover name="InAn" scorefxn="REF2015" packstat="1" tracer="0"       \
            pack_input="0" interface_sc="1" pack_separated="1" interface="A_BC" />
        <AlignChain name="align" source_chain="0" target_chain="0" target_name="Native.pdb"  \
            align_to_com="0" />
        <RunSimpleMetrics name="RMSD" metrics="rmsd" prefix="rmsd_" />
    </MOVERS>
    <PROTOCOLS>
        <Add mover="repack" />
        <Add mover="minimize_sc" />
        <Add mover="Minim" />
        <Add mover="FR" />
        <Add mover="align" />
        <Add mover="RMSD" />
        <Add mover="InAn" />
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

Protocols example

```
<ROSETTASCRIPTS>
    ...
    <MOVERS>
1     <PackRotamersMover name="repack" scorefxn="REF2015" task_operations="ifcl,rtr" />
      <MinMover name="minimize_sc" scorefxn="REF2015" chi="1" bb="0" tolerance="0.001"     \
2         jump="0" type="dfpmin_armijo_nonmonotone" />
      <MinMover name="Minim" jump="ALL" cartesian="0" bondangle="0" omega="0"             \
3         bondlength="0" chi="1" bb="1" />
4     <FastRelax name="FR" scorefxn="REF2015" disable_design="1" task_operations="ifcl"/>
      <InterfaceAnalyzerMover name="InAn" scorefxn="REF2015" packstat="1" tracer="0"      \
5         pack_input="0" interface_sc="1" pack_separated="1" interface="A_BC" />
6     <AlignChain name="align" source_chain="0" target_chain="0" target_name="Native.pdb"  \
          align_to_com="0" />
7     <RunSimpleMetrics name="RMSD" metrics="rmsd" prefix="rmsd_" />
    </MOVERS>
    <PROTOCOLS>
        <Add mover="repack" />
        <Add mover="minimize_sc" />
        <Add mover="Minim" />
        <Add mover="FR" />
        <Add mover="align" />
        <Add mover="RMSD" />
        <Add mover="InAn" />
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

Protocols example

```
<ROSETTASCRIPTS>
    ...
    <MOVERS>
      <RunSimpleMetrics name="RMSD" metrics="rmsd" prefix="rmsd_" />
      <AlignChain name="align" source_chain="0" target_chain="0" target_name="Native.pdb"  \
          align_to_com="0" />
      <PackRotamersMover name="repack" scorefxn="REF2015" task_operations="ifcl,rtr" />
      <MinMover name="minimize_sc" scorefxn="REF2015" chi="1" bb="0" tolerance="0.001"     \
          jump="0" type="dfpmin_armijo_nonmonotone" />
      <FastRelax name="FR" scorefxn="REF2015" disable_design="1" task_operations="ifcl"/>
      <InterfaceAnalyzerMover name="InAn" scorefxn="REF2015" packstat="1" tracer="0"       \
          pack_input="0" interface_sc="1" pack_separated="1" interface="A_BC" />
      <MinMover name="Minim" jump="ALL" cartesian="0" bondangle="0" omega="0"              \
          bondlength="0" chi="1" bb="1" />
    </MOVERS>
    <PROTOCOLS>
        <Add mover="repack" />
        <Add mover="minimize_sc" />
        <Add mover="Minim" />
        <Add mover="FR" />
        <Add mover="align" />
        <Add mover="RMSD" />
        <Add mover="InAn" />
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

6
5
1
2
4
7
3
1
2
3
4
5
6
7

Protocols example

```
<ROSETTASCRIPTS>
    ...
    <MOVERS>
      <RunSimpleMetrics name="RMSD" metrics="rmsd" prefix="rmsd_" />
      <AlignChain name="align" source_chain="0" target_chain="0" target_name="Native.pdb"  \
          align_to_com="0" />
      <PackRotamersMover name="repack" scorefxn="REF2015" task_operations="ifcl,rtr" />
      <MinMover name="minimize_sc" scorefxn="REF2015" chi="1" bb="0" tolerance="0.001"     \
          jump="0" type="dfpmin_armijo_nonmonotone" />
      <FastRelax name="FR" scorefxn="REF2015" disable_design="1" task_operations="ifcl"/>
      <InterfaceAnalyzerMover name="InAn" scorefxn="REF2015" packstat="1" tracer="0"       \
          pack_input="0" interface_sc="1" pack_separated="1" interface="A_BC" />
      <MinMover name="Minim" jump="ALL" cartesian="0" bondangle="0" omega="0"              \
          bondlength="0" chi="1" bb="1" />
    </MOVERS>
    <PROTOCOLS>
        <Add mover="repack" />
        <Add mover="Minim" />
        <Add mover="RMSD" />
        <Add mover="InAn" />
    </PROTOCOLS>
</ROSETTASCRIPTS>
```

3 - 1 2 - 4 -

And other tags

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

- Packer Palette
- Simple Metrics
- Apply To Pose
- Outputs

https://docs.rosettacommons.org/docs/latest/scripting_documentation/RosettaScripts/RosettaScripts#rosettascript-sections_protocols

Useful Features

- Rewrite old Rosetta XML scripts:

.../tools/xsd_xrw/rewrite_rosetta_scripts.py

- Validate your XML scripts

https://docs.rosettacommons.org/docs/latest/application_documentation/rosetta_scripts/validate_rosetta_script

- Variable substitutions

.../rosetta_scripts.default.linuxgccrelease **-parser:script_vars** **resfile=A105T.resfile** -parser:protocol design.xml ...

```
<ROSETTASCRIPTS>
            <TASKOPERATIONS>
                        <InitializeFromCommandline name="ifcl" />
                        <ReadResfile name="rrf" filename="%%resfile%%"/>
            </TASKOPERATIONS>
            <MOVERS>
                        <PackRotamersMover name="design" \
                        scorefxn="ref2015" task_operations="ifcl,rrf" />
            </MOVERS>
            <PROTOCOLS>
                        <Add mover="design" />
            </PROTOCOLS>
            <OUTPUT scorefxn="ref2015" />
</ROSETTASCRIPTS>
```

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

References

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