MakeBundle

Search... Home Feedback

Getting Started

Build Documentation

Rosetta Basics

Application Documentation

Scripting Interfaces

- RosettaScripts
  - Movers
  - Residue Selectors
  - PackerPalettes
  - Task Operations
  - Simple Metrics
  - Filters
  - FeaturesReporters
  - Composite Protocols
  - Databases with RosettaScripts
  - Formatting conventions
  - For Developers
- TopologyBroker
- PyRosetta

Development Documentation

FAQ

Glossary

Options list

Page created by Vikram K. Mulligan (vmulligan@flatironinstitute.org). Last modified 12 October 2018. *Back to Mover page.*

Table of Contents

MakeBundle

Generates a helical bundle using the Crick equations (which describe a helix of helices) or modified Crick equations (describing a laterally-squashed helix of helices). This mover is general enough to create arbitrary helices using arbitrary backbones. Since strands are a special case of a helix (in which the turn per residue is about 180 degrees), the mover can also generate beta-barrels or other strand bundles. The generated secondary structure elements are disconnected, so subsequent movers (e.g. **GeneralizedKIC**) must be invoked to connect them with loops. Parameters are stored with the pose, and are written in REMARK lines on PDB output.

Helix types are defined with crick_params files, located in the Rosetta database in database/protocol_data/crick_parameters (or provided by the user). Support for Crick parameter files defining helices in which the repeating unit is more than one residue has recently been added. For more information on this file type, see this page.

Full options

Autogenerated Tag Syntax Documentation:

The MakeBundle mover builds a helical bundle parametrically, using the Crick parameterization, given a set of Crick parameter values. Note that the Crick parameterization is compatible with arbitrary helices (including strands, which are special cases of helices in which the turn per residue is about 180 degrees).

```
<MakeBundle name="(&string;)" r0="(0.000000 &real;)" omega0="(0.000000 &real;
        delta_omega0="(0.000000 &real;)" delta_omega1="(0.000000 &real;)"
        delta_t="(0.000000 &real;)" z0_offset="(0.000000 &real;)"
        z1_offset="(0.000000 &real;)" epsilon="(1.000000 &real;)"
        repeating_unit_offset="(0 &non_negative_integer;)"
        r1_peratom="(0 &real_wsslist;)" omega1="(0.000000 &real;)"
        z1="(0.000000 &real;)" delta_omega1_peratom="(0 &real_wsslist;)"
        delta_z1_peratom="(0 &real_wsslist;)" invert="(false &bool;)"
        set_dihedrals="(true &bool;)" set_bondangles="(true &bool;)"
        set_bondlengths="(true &bool;)" use_degrees="(false &bool;)"
        symmetry="(0 &non_negative_integer;)"
        symmetry_copies="(0 &non_negative_integer;)" reset="(false &bool;)"
        crick_params_file="(&string;)" residue_name="(&string;)"
        helix_length="(0 &non_negative_integer;)" >
    <Helix crick_params_file="(&string;)" residue_name="(&string;)"
            helix_length="(0 &non_negative_integer;)" r0="(0.000000 &real;)"
            r0_copies_helix="(0 &non_negative_integer;)" omega0="(0.000000 &r
            omega0_copies_helix="(0 &non_negative_integer;)"
            pitch_from_helix="(0 &non_negative_integer;)"
            delta_omega0="(0.000000 &real;)"
            delta_omega0_copies_helix="(0 &non_negative_integer;)"
            delta_omega1="(0.000000 &real;)"
            delta_omega1_copies_helix="(0 &non_negative_integer;)"
            delta_t="(0.000000 &real;)"
            delta_t_copies_helix="(0 &non_negative_integer;)"
            z0_offset="(0.000000 &real;)"
            z0_offset_copies_helix="(0 &non_negative_integer;)"
            z1_offset="(0.000000 &real;)"
            z1_offset_copies_helix="(0 &non_negative_integer;)"
            epsilon="(1.000000 &real;)"
            epsilon_copies_helix="(0 &non_negative_integer;)"
            repeating_unit_offset="(0 &non_negative_integer;)"
            r1_peratom="(0 &real_wsslist;)" omega1="(0.000000 &real;)"
            omega1_copies_helix="(0 &non_negative_integer;)" z1="(0.000000 &r
            z1_copies_helix="(0 &non_negative_integer;)"
            delta_omega1_peratom="(0 &real_wsslist;)"
            delta_z1_peratom="(0 &real_wsslist;)" invert="(false &bool;)"
            set_dihedrals="(true &bool;)" set_bondangles="(true &bool;)"
            set_bondlengths="(true &bool;)" />
</MakeBundle>
```

- **r0**: Major helix radius, in Angstroms.
- **omega0**: Major helix twist per residue, stored in radians.
- **delta_omega0**: Rotation of a helix about the z-axis, stored in radians.
- **delta_omega1**: Rotation of a helix about its own axis, stored in radians.
- **delta_t**: Offset along the polypeptide backbone, in residues.
- **z0_offset**: Offset along the global z-axis, in Angstroms.
- **z1_offset**: Offset along the superhelical path through space, in Angstroms.

- **epsilon**: Lateral squash parameter/eccentricity of the cross-section of a bundle or barrel.
- **repeating_unit_offset**: Shift, in residues, of the repeating unit of a helix.
- **r1_peratom**: Minor helix radius -- a vector of real numbers in Angstroms, with one per atom in the repeating unit of a helix. Read from Crick params file, and not normally set by hand.
- **omega1**: Minor helix twist per residue, stored in radians. Read from Crick params file, and not normally set by hand, sampled, or perturbed.
- **z1**: Minor helix rise per residue along the helix axis, in Angstroms. Read from Crick params file, and not normally set by hand, sampled, or perturbed.
- **delta_omega1_peratom**: Minor helix angular offsets of each mainchain atom in the repeating unit, in radians. Read from Crick params file, and not normally set by hand.
- **delta_z1_peratom**: Minor helix axial offsets of each mainchain atom in the repeating unit, in Angstroms. Read from Crick params file, and not normally set by hand.
- **invert**: Inversion state of this helix -- true for inverted.
- **set_dihedrals**: True indicates that the parametric machinery will set mainchain torsion values.
- **set_bondangles**: True indicates that the parametric machinery will allow mainchain bond angle values to deviate from ideality.
- **set_bondlengths**: True indicates that the parametric machinery will allow mainchain bond length values to deviate from ideality.
- **use_degrees**: Input values in degrees, instead of radians
- **symmetry**: Symmetry setting (n-fold; 0 or 1 === no symmetry
- **symmetry_copies**: How many symmetry copies will be generated? 'All' if zero, only the first one if 1, but you can ask for any other number
- **reset**: Reset the input pose, instead of appending the bundle to it
- **crick_params_file**: File name of a file containing Crick parameters for the secondary structure type desired.
- **residue_name**: For a specific helix, residue, indicated by name, from which to build the helical bundle.
- **helix_length**: For a specific helix, length, in residues, for this helix

Subtag **Helix**:

- **crick_params_file**: File name of a file containing Crick parameters for the secondary structure type desired.
- **residue_name**: For a specific helix, residue, indicated by name, from which to build the helical bundle.
- **helix_length**: For a specific helix, length, in residues, for this helix
- **r0**: Major helix radius, in Angstroms.
- **r0_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for r0 should be copied.
- **omega0**: Major helix twist per residue, stored in radians.
- **omega0_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for omega0 should be copied.
- **pitch_from_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which pitch value should be copied in order to set omega0, the twist per residue. An alternative to "omega0_copies_helix".
- **delta_omega0**: Rotation of a helix about the z-axis, stored in radians.
- **delta_omega0_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for delta_omega0 should be copied.
- **delta_omega1**: Rotation of a helix about its own axis, stored in radians.
- **delta_omega1_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for delta_omega1 should be copied.
- **delta_t**: Offset along the polypeptide backbone, in residues.
- **delta_t_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for delta_t should be copied.
- **z0_offset**: Offset along the global z-axis, in Angstroms.
- **z0_offset_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for z0_offset should be copied.
- **z1_offset**: Offset along the superhelical path through space, in Angstroms.
- **z1_offset_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for z1_offset should be copied.
- **epsilon**: Lateral squash parameter/eccentricity of the cross-section of a bundle or barrel.
- **epsilon_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for epsilon should be copied.
- **repeating_unit_offset**: Shift, in residues, of the repeating unit of a helix.
- **r1_peratom**: Minor helix radius -- a vector of real numbers in Angstroms, with one per atom in the repeating unit of a helix. Read from Crick params file, and not normally set by hand.
- **omega1**: Minor helix twist per residue, stored in radians. Read from Crick params file, and not normally set by hand, sampled, or perturbed.
- **omega1_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for omega1 should be copied.
- **z1**: Minor helix rise per residue along the helix axis, in Angstroms. Read from Crick params file, and not normally set by hand, sampled, or perturbed.

- **z1_copies_helix**: The index of the parametric object (e.g. the helix, in the case of a helical bundle) from which the value for z1 should be copied.
- **delta_omega1_peratom**: Minor helix angular offsets of each mainchain atom in the repeating unit, in radians. Read from Crick params file, and not normally set by hand.
- **delta_z1_peratom**: Minor helix axial offsets of each mainchain atom in the repeating unit, in Angstroms. Read from Crick params file, and not normally set by hand.
- **invert**: Inversion state of this helix -- true for inverted.
- **set_dihedrals**: True indicates that the parametric machinery will set mainchain torsion values.
- **set_bond**angles: True indicates that the parametric machinery will allow mainchain bond angle values to deviate from ideality.
- **set_bond**lengths: True indicates that the parametric machinery will allow mainchain bond length values to deviate from ideality.

Crick params files

The Rosetta database currently contains several sets of minor helix parameters:

- "alpha_helix": A standard L-amino acid right-handed alpha-helix, with phi=-64.8, psi=-41.0, and omega=180.0. Note that the turn per residue for this helix is 98.65 degrees, not exactly 100 degrees.
- "alpha_helix_100": An L-amino acid right-handed alpha-helix, with phi=-62.648, psi=-41.0, and omega=180.0, yielding a turn per residue of 100 degrees. This is for backward compatibility with the Python scripts used to generate helical bundles previously.
- "beta_strand": An L-amino acid beta-strand, with phi=-135.0, psi=135.0, and omega=180.0.
- "collagen": **EXPERIMENTAL**. This generates a collagen helix. This is experimental because the repeating unit in this helix is a three-residue repeat rather than a single residue.
- "neutral_beta_strand": An unnaturally straight beta-strand, with phi=180.0, psi=180.0, and omega=180.0. Both L- and D-amino acids can access this region of Ramachandran space.
- "L_alpha_helix": A left-handed alpha-helix, as can be formed by D-amino acids. Phi=64.8, psi=41.0, and omega=180.0.
- "daa_beta_strand": A beta-strand formed by D-amino acids, mirroring that formed by L-amino acids. Phi=135.0, psi=-135.0, and omega=180.0.
- "14_helix": A left-handed helix formed by beta-amino acids. Phi=-139.9, theta=59.5, psi=-138.7, and omega=180.0.

Example

This script generates an antiparallel beta-barrel with a bundle of alpha-helices on the inside.

```
<MakeBundle name="bundle1" set_bondlengths="true" set_bondangles="true" resid
        #The parameters set above ensure that by default, each "helix" will a
        <Helix /> #A strand
        <Helix delta_omega0="0.19634954" invert="1" delta_t="0.25" delta_omeg
        <Helix r0="21" omega0="0.05" crick_params_file="alpha_helix" helix_le
        #The three elements defined above are repeated 16 times about the bun
</MakeBundle>
```

Note that RosettaScripts requires some sort of input on which to operate, but this mover, by default, deletes input geometry and replaces it with the generated geometry. When running RosettaScripts, one can either pass in a dummy PDB file with the -in:file:s flag, or a dummy FASTA file with the -in:file:fasta flag.

See Also

- The Crick params file format
- BundleGridSampler mover
- PerturbBundle mover
- BundleReporter filter
- Symmetry: Using symmetry in Rosetta
- SymmetryAndRosettaScripts
- SetupForSymmetryMover
- SetupNCSMover
- DetectSymmetryMover
- SymMinMover
- SymPackRotamersMover
- ExtractAsymmetricUnitMover
- ExtractAsymmetricPoseMover