
Mollib Documentation

Release 1.0.7

Justin L Lorieau

Sep 03, 2017

Contents

1	Command Line Interface	2
1.1	Overview	2
1.2	process command	4
1.3	measure command	5
1.4	hbonds command	13
1.5	pa command	16
1.6	Configuration Options	22
2	Recipes	31
2.1	Command-line Tips and Tricks	31
2.2	Python Recipes	31
3	Releases	33
3.1	Mollib 1.0 release notes	33
4	Development	36
4.1	Testing	37
4.2	Makefile Options	37
4.3	Including Datasets	38
4.4	Adding Plugins	38
4.5	Branches	38
4.6	Docstring Format	38
5	Building and Deployment	42
5.1	Building	42
5.2	Deployment	43
6	API Documentation	44
6.1	Core Module	44
6.2	Plugins Module	46
6.3	Hydrogens Module	46
6.4	Hbonds Module	46
6.5	Partial Alignment Module	51
6.6	Statistics Module	53
6.7	Utilities	53
	Bibliography	55

Mollob is a *unified* command-line program and Python library for the validation, quality analysis and manipulation of molecular structures with an emphasis on biophysical analysis. Mollob is built on a plugin framework to easily add new tools to manipulate and analyze structures and data, which can then be *combined* and *cross-validated*.

Mollob includes tools for:

- The *processing* and protonation of molecules.
- The analysis of *hydrogen bonds* and hydrogen bond quality compared to the highest-resolution PDB structures. Hydrogen bonds are classified based on their acceptor-donor residues and Ramachandran angles
- The *measurement* of geometries within molecules as well as the reporting and classification of Ramachandran angles.
- The statistical analysis and comparison of structures to high-resolution crystal structures.
- The analysis of *partial alignment* data with residual dipolar coupling (RDC) and residual anisotropic chemical shift (RACS, a.k.a RCSA) data.

Installation

```
$ pip install mollob
```

Note: Python and pip may not be installed. You may need to [download](#) and install Python first.

Command Line Interface

Overview

The command line interface includes all of mollib's functions for processing and analyzing molecules and molecular data. Mollib can be accessed either through the mollib or ml commands.

```
$ ml --help
usage: mollib [-h] [-d | -s | -v] [--list-plugins] [--list-settings]
              [--version]
              ...

A molecular processor

optional arguments:
  -h, --help            show this help message and exit
  -d, --debug           Print debugging statements
  -s, --suppress        Suppress all messages, except critical
  -v, --verbose         Print extra information
  --list-plugins        List the installed plugins
  --list-settings       List the available sections for settings
  --version             Show the program version

commands:

  process              Process the structure
  measure              Measure geometries in molecules
  hbonds               Find and report hydrogen bonds in molecules
  pa                   Singular Value Decomposition of partially aligned NMR data
                      (RDCs and RACsSs)
```

-h / --help Get basic help information on usage of the command line interface, or one one of the mollib commands.

-d / --debug Display debug messages to the terminal. This is usually only useful to developers or in helping to debug issues.

-s / --suppress Display only errors and critical errors.

-v / --verbose Display informational messages to the terminal. By default, only warning and error messages are displayed.

--version Display the version number of the installed mollib.

--list-plugins Display a list of installed and enabled plugins.

```
$ ml --list-plugins
Installed plugins:
  process      Enabled
  measure      Enabled
  hydrogenate   Enabled
  hbonds       Enabled
  pa           Enabled
```

--list-settings Display a list of the settings sections that will be interpreted from configuration files.

```
$ ml --list-settings
Installed settings sections:
  [settings]
  [utils.settings]
  [hydrogens.settings]
  [hbonds.settings]
  [statistics.settings]
  [pa.settings]
```

Configuration Files

Mollib will check for a `.molllibrc` configuration file and a configuration file passed with the `-c filename / --config filename` commandline argument for configuration parameters.

Configuration files customize mollib's behavior from default values. If a `.molllibrc` file is present and a configuration file is specified in the commandline, the commandline configuration file will take precedence for parameters that are mentioned in both.

The following is an example configuration file. Note that only the parameters that need to be changed can be specified in the file.

Listing 1.1: setup.cfg

```
[settings]
default_pH = 7.0
pKs = {'ASP': {'OD1-OD2': (-1.0, 3.5)},
      'GLU': {'OE1-OE2': (-1.0, 4.2)},
      'HIS': {'ND1-NE2': (6.6, 14.0)},
      'CYS': {'SG': (6.8,)},
      'TYR': {'OH': (10.3,)},
      'LYS': {'NZ': (10.5, 14.0, 14.0)},
      'last': {'O-OXT': (-1.0, 3.3)},
      'first': {'N': (7.7, 14.0, 14.0)},
      }

[settings.hydrogens]
bond_length = {'N-H': 1.023,
               'C-H': 1.117,
```

```
'C-C': 1.517,
'N-H2': 1.01,
'O-H': 0.958,
'C-N': 1.488,
'C=O': 1.250,
'C-O': 1.413,
}
```

process command

The process command is the main command for processing, reading and writing files. All of the options and preprocessors available in process are available to other commands.

```
$ ml process --help
usage: mollob process [-h] -i id/filename [id/filename ...] [-c filename] [-l]
                    [-s] [-m [MODELS [MODELS ...]]]
                    [-o [filename [filename ...]]] [--hydrogenate]

arguments:
  -h, --help            show this help message and exit
  -i id/filename [id/filename ...], --in id/filename [id/filename ...]
                        (required) The filename(s) or PDB identifier(s) of the
                        structure(s)
  -c filename, --config filename
                        The configuration filename
  -l                    List details on the molecule(s)
  -s, --save            Save fetched files to the local directory.
  -m [MODELS [MODELS ...]], --models [MODELS [MODELS ...]]
                        The models numbers to analyze.
  -o [filename [filename ...]], --out [filename [filename ...]]
                        The output filename(s) for the structure(s)
  --hydrogenate         Strip hydrogens and re-add them before analysis
```

Arguments

-i / --in id/filename (required) One or more structural identifiers (ex: PDB file identifiers) or filenames to load.

If the structure could not be found locally, a copy will be downloaded and cached for further analysis.

Multiple input identifiers and filenames can be used to process multiple files simultaneously.

-o / --out filename The output filename(s).

Structure files that are written have passed through the mollob parser and will likely not be identical to the original files. Changes may have been done to the atoms, header comments or other aspects, depending on which preprocessors were used.

Multiple output filenames can be used, and these will be matched to the corresponding entries in the input filenames or identifiers.

-c / --config filename The configuration file. See [Configuration files](#) for more details.

-s / --save Save fetched files from the internet to the local directory.

-m / --models The specific model numbers to load. By default, only the first model is loaded.

- l List details on the molecule, including the number of chains, residues and atoms. This option is helpful in determining selections in *Abbreviated Selectors*.

Preprocessors Arguments

- hydrogenate Strip all hydrogen atoms and re-add hydrogens based on ideal geometry. More details on the current implementation can be found in the API documentation *Hydrogens Module*.

Note: Adding two hydrogens to an sp² heavy atom will label the E-hydrogen ‘H1’ and the Z-hydrogen ‘H2’. This situation happens with the HD2 hydrogens of asparagine residues, for example.

Note: Adding two hydrogens to an sp³ heavy atom will label the pro-R hydrogen ‘H2’ and the pro-S hydrogen ‘H3’. Some exceptions in proteins exist. By default, the JBNMR 12, 1-23 (1998) convention is followed.

Examples

The following example loads a crystal structure of ubiquitin from the PDB (-i 1UBQ), adds hydrogens to the molecule (--hydrogenate) and saves the output to a new file (-o 1UBQ_H.pdb).

```
$ ml process -i 1UBQ -o 1UBQ_H.pdb --hydrogenate
```

measure command

The measure command is used for measuring geometries in molecules. All of the options and preprocessors available from the *process command* are also available.

```
$ ml measure --help
usage: molLib measure [-h] -i id/filename [id/filename ...] [-c filename] [-l]
                        [-s] [-m [MODELS [MODELS ...]]]
                        [-d atom atom | -a atom atom atom | -dih atom atom atom atom | -w atom atom]
                        [--stats] [--only-intra] [--exclude-intra]
                        [--only-intra-chain] [--exclude-intra-chain]
                        [--only-delta DELTA] [--only-bonded] [--hydrogenate]
                        [--rama]

arguments:
  -h, --help            show this help message and exit
  -i id/filename [id/filename ...], --in id/filename [id/filename ...]
                        (required) The filename(s) or PDB identifier(s) of the
                        structure(s)
  -c filename, --config filename
                        The configuration filename
  -l                    List details on the molecule(s)
  -s, --save            Save fetched files to the local directory.
  -m [MODELS [MODELS ...]], --models [MODELS [MODELS ...]]
                        The models numbers to analyze.
  --hydrogenate         Strip hydrogens and re-add them before analysis
```

```

measurement options:
  -d atom atom, --dist atom atom
                                Measure distances between 2 atom selections. ex: 31.N
                                32.CA
  -a atom atom atom, --angle atom atom atom
                                Measure angles between 3 atom selections. ex: 31.N
                                31.CA 31.C
  -dih atom atom atom atom, --dihedral atom atom atom atom
                                Measure dihedral angles between 4 atom selections. ex:
                                31.N 31.CA 31.C 32.N
  -w atom atom, --within atom atom
                                Measure all distances from atom selection to within
                                the specified distance. ex: 31:33.N 5
  --stats                       Report statistics on the reported measurements.
  --rama                       Report the Ramachandran angles. Filters and options
                                are ignored.

```

```

filters:
  --only-intra                 Only report measurements within a residue
  --exclude-intra              Exclude measurements within a residue
  --only-intra-chain           Only report measurements within a chain
  --exclude-intra-chain        Exclude measurements within a chain
  --only-delta DELTA           Only report residues separated by DELTA residue
                                numbers
  --only-bonded                Only report measurements from bonded atoms

```

Arguments

-d / --dist atom atom Measure the distance (in Angstroms) between two atoms.

Multiple atom pairs can be used. ex: -d 31.N 31.CA -d 32.N 33.CA

Atoms must follow the standard naming conventions. See [Abbreviated Selectors](#) and [Filters](#).

-a / --angle Measure the angle (in degrees) between three atoms.

Multiple atom triplets can be used. ex: -a 31.N 31.CA 31.CB -a 32.N 32.CA 32.CB

Atoms must follow the standard naming conventions. See [Abbreviated Selectors](#) and [Filters](#).

-dih / --dihedral Measure the dihedral angle (in degrees) between four atoms.

Multiple atom quartets can be used. ex: -dih 30.C 31.N 31.CA 31.C -dih 31.N 31.CA 31.C 32.N

Atoms must follow the standard naming conventions. See [Abbreviated Selectors](#) and [Filters](#).

Note: If simple Ramachandran dihedrals are needed, checkout `--rama`.

--stats Report the average and standard deviation of all measured values. This option only applies to the distance, angle and dihedral measurements.

--rama Measure Ramachandran angles (in degrees) for a protein. Filters and options are ignored. Heteroatom chains are skipped.

The `--rama` command classifies Ramachandran angles based on backbone-backbone amide hydrogen bonds.

Atom Selectors and Filters

Abbreviated Selectors

The measure methods find atoms using atom locators. Atom locators must follow one of these conventions:

1. (residue number).(atom name). ex: 31.CB for the CB atom of residue number 31.
2. (chain id).(residue number).(atom name). ex: A.31.CB for the CB atom of residue number 31 in chain 'A'.

Additionally, the chain id, residue number or both can be expressed as a range using the ':' character:

1. (residue range).(atom name). ex: 31:34.CB for the CB atom of residue number 31, 32, 33 and 34.
2. (chain range).(residue number).(atom name). ex:A:C.34.CB for the CB atom of residue number 34 for chains 'A', 'B', 'C' and 'D'.

Finally, heteroatom chains have an asterisk appended to them. ex: 'C*'

Note: Atom selections may encompass hundreds of atoms, which when used in combination, could lead to searches over millions of combinations. To help improve their performance, you can either narrow their scope by reducing the range of chains or residue numbers, combine multiple *Filters* or use one of the shortcut selectors, like `--rama` for Ramachandran dihedral angles.

Filters

--only-intra Exclude atom selections that are *not* within the same residue.

--exclude-intra Exclude atom selections that are within the same residue.

--only-intra-chain Exclude atom selections that are *not* within the same chain.

--exclude-intra-chain Exclude atom selections that are within the same chain.

--only-delta DELTA Exclude atom selections that don't have at least one set of atoms with residues separated by DELTA number. This filter ignores the chain identifier and may need to be combined with `--filter-intra-chain` or `--exclude-intra-chain`.

--only-bonded Exclude atom selections that are not bonded. The bonded tests linear bonding relationships. For example, a dihedral with four atoms (atom1, atom2, atom3 and atom4) must have bonds between atom1–atom2, atom2–atom3 and atom3–atom4.

Note: Bonded searches have to investigate the topology of each atom selection, which can be slower than the above filters. Combining the `--only-bonded` filter with other filters, like `--only-delta 1`, can significantly speed up searches.

Examples

Measuring distances

Measure α -helical HA-H distances in chain 'A' for residues 23-49 of 2MUV, the homotetrametic influenza M2 channel. Include statistics on the measured distances.

```
$ ml measure -i 2MUV -d 23:49.HA 23:49.H --only-delta 3 --stats
```

Table: Distances for 2MUV-1

Num	Atom 1	Atom 2	Dist. (A)

1	A.S23.HA	A.L26.H	7.01
2	A.D24.HA	A.V27.H	5.08
3	A.P25.HA	A.V28.H	3.38
4	A.L26.HA	A.A29.H	3.58
5	A.V27.HA	A.A30.H	3.64
6	A.V28.HA	A.N31.H	3.58
7	A.A29.HA	A.I32.H	3.42
8	A.A30.HA	A.I33.H	3.59
9	A.N31.HA	A.G34.H	3.53
10	A.I32.HA	A.I35.H	3.75
11	A.I33.HA	A.L36.H	3.42
12	A.I35.HA	A.L38.H	3.36
13	A.L36.HA	A.I39.H	3.54
14	A.H37.HA	A.L40.H	3.49
15	A.L38.HA	A.W41.H	3.42
16	A.I39.HA	A.I42.H	3.77
17	A.L40.HA	A.L43.H	3.63
18	A.W41.HA	A.D44.H	3.46
19	A.I42.HA	A.R45.H	3.61
20	A.L43.HA	A.L46.H	3.36
21	A.D44.HA	A.F47.H	3.58
22	A.R45.HA	A.F48.H	3.67
23	A.L46.HA	A.K49.H	6.94

			4.0 ± 1.0

Measure CA-CA distances between residue 20-21 for chains 'A', 'B', 'C' and 'D' of 2MUV—excluding same residue distances and same chain distances.

```
$ ml measure -i 2MUV -d A:D.20:21.CA A:D.20:21.CA --exclude-intra --exclude-intra-chain
```

Table: Distances for 2MUV-1

Num	Atom 1	Atom 2	Dist. (A)

1	A.N20.CA	B.N20.CA	18.05
2	A.N20.CA	B.D21.CA	17.50
3	A.D21.CA	B.N20.CA	14.39
4	A.D21.CA	B.D21.CA	13.70
5	A.N20.CA	C.N20.CA	25.10
6	A.N20.CA	C.D21.CA	21.94
7	A.D21.CA	C.N20.CA	21.95
8	A.D21.CA	C.D21.CA	19.02
9	A.N20.CA	D.N20.CA	17.86
10	A.N20.CA	D.D21.CA	14.21
11	A.D21.CA	D.N20.CA	17.32
12	A.D21.CA	D.D21.CA	13.52
13	B.N20.CA	C.N20.CA	17.89
14	B.N20.CA	C.D21.CA	17.44
15	B.D21.CA	C.N20.CA	14.21
16	B.D21.CA	C.D21.CA	13.64
17	B.N20.CA	D.N20.CA	25.32

```

18  B.N20.CA  D.D21.CA  22.15
19  B.D21.CA  D.N20.CA  22.14
20  B.D21.CA  D.D21.CA  19.21
21  C.N20.CA  D.N20.CA  17.51
22  C.N20.CA  D.D21.CA  17.00
23  C.D21.CA  D.N20.CA  13.86
24  C.D21.CA  D.D21.CA  13.20

```

Compare the distance between the HA of residue 5 and the H of residue 21 for two different structures, 2KXA and 2LWA. The 2KXA structure represents the wildtype hemagglutinin fusion peptide (HAfp) in the *closed* helical-hairpin structure, placing these two atoms in close proximity. The 2LWA structure represents the conformational ensemble of the HAfp-G8A mutant with a closed structure (chain ‘A’), a semi-closed structure (chain ‘B’) and an open structure (chain ‘C’).

```
$ ml measure -i 2KXA 2LWA -d A:C.5.HA A:C.21.H --only-intra-chain
```

Table: Distances for 2KXA-1

Num	Atom 1	Atom 2	Dist. (A)
1	A.A5.HA	A.W21.H	3.30

Table: Distances for 2LWA-1

Num	Atom 1	Atom 2	Dist. (A)
1	A.A5.HA	A.W21.H	3.22
2	B.A5.HA	B.W21.H	11.77
3	C.A5.HA	C.W21.H	18.47

Measuring Angles

Measure the angle of the bonded ‘C-I’–‘N’–‘H’ atoms for residues 20-30 from the ubiquitin structure 2MJB.

```
$ ml measure -i 2MJB -a 20:30.C 20:30.N 20:30.H --only-bonded
```

Table: Angles for 2MJB-1

Num	Atom 1	Atom 2	Atom 3	Angle (deg)
1	A.S20.C	A.D21.N	A.D21.H	120.2
2	A.D21.C	A.T22.N	A.T22.H	118.4
3	A.T22.C	A.I23.N	A.I23.H	119.0
4	A.I23.C	A.E24.N	A.E24.H	118.7
5	A.E24.C	A.N25.N	A.N25.H	119.2
6	A.N25.C	A.V26.N	A.V26.H	118.9
7	A.V26.C	A.K27.N	A.K27.H	119.2
8	A.K27.C	A.A28.N	A.A28.H	118.5
9	A.A28.C	A.K29.N	A.K29.H	118.4
10	A.K29.C	A.I30.N	A.I30.H	118.9

Measuring Dihedrals

The following example measures the ϕ angle for residues 2-6 of the hemagglutinin fusion peptide domain (2KXA).

```
$ ml measure -i 2KXA -dih 2:6.C 2:6.N 2:6.CA 2:6.C --only-bonded --stats
```

Table: Dihedrals for 2KXA-1

Num	Atom 1	Atom 2	Atom 3	Atom 4	Dihedral (deg)
1	A.L2.C	A.F3.N	A.F3.CA	A.F3.C	-65.2
2	A.F3.C	A.G4.N	A.G4.CA	A.G4.C	-57.2
3	A.G4.C	A.A5.N	A.A5.CA	A.A5.C	-68.4
4	A.A5.C	A.I6.N	A.I6.CA	A.I6.C	-61.7

					-63.0 ± 4.0

Ramachandran Angles

Measure the Ramachandran ϕ and ψ angles for the hemagglutinin fusion peptide structure 2KXA.

```
$ ml measure -i 2KXA --rama
```

Table: Ramachandran angles for 2KXA-1

Residue	Phi (deg)	Psi (deg)	Classification	Minor	E (kT) / Prob.
A.G1	0.0	-179.9			0.0 / 100.0%
A.L2	-61.0	-45.6	alpha-helix	N-term	1.1 / 31.9%
A.F3	-65.2	-48.3	alpha-helix		1.7 / 18.4%
A.G4	-57.2	-32.3	alpha-helix		1.9 / 15.4%
A.A5	-68.4	-45.0	alpha-helix		0.7 / 51.0%
A.I6	-61.7	-51.1	alpha-helix		2.5 / 8.1%
A.A7	-66.7	-42.8	alpha-helix		0.7 / 51.0%
A.G8	-63.4	-34.5	alpha-helix		1.4 / 23.5%
A.F9	-65.4	-44.3	alpha-helix		0.7 / 51.0%
A.I10	-65.1	-28.7	alpha-helix		3.4 / 3.4%
A.E11	-92.2	-47.2	alpha-helix		7.5 / 0.1%
A.G12	-111.9	28.0	alpha-helix	C-term	5.1 / 0.6%
A.G13	44.9	-146.5			7.2 / 0.1%
A.W14	-49.7	-61.9	alpha-helix	N-term	5.7 / 0.3%
A.T15	-49.6	-32.2	alpha-helix		8.4 / 0.0%
A.G16	-69.5	-37.4	alpha-helix		0.9 / 38.9%
A.M17	-58.3	-46.5	alpha-helix		0.8 / 45.5%
A.I18	-62.6	-49.1	alpha-helix		0.6 / 54.7%
A.D19	-55.5	-45.6	alpha-helix		0.8 / 45.5%
A.G20	-65.7	-34.8	alpha-helix		1.7 / 17.7%
A.W21	-62.9	-48.0	alpha-helix		0.6 / 54.7%
A.Y22	-77.6	-33.0	alpha-helix	C-term	2.9 / 5.3%
A.G23	74.3	-87.9			6.7 / 0.1%
A.S24	-56.7	0.0			0.0 / 100.0%

Approach to Secondary Structure Assignments

β Turns

Turns are defined by a hydrogen bond between residues ‘i’ and ‘i+4’ as well as the backbone torsion angles for residues ‘i+1’ and ‘i+2’. The turn type is based on the torsion angles of the ‘i+1’ and ‘i+2’ residues.

Type	ϕ_{i+1}	ψ_{i+1}	ϕ_{i+2}	ψ_{i+2}
I	-60°	-30°	-90°	0°
I'	60°	30°	90°	0°
II	-60°	120°	80°	0°
II'	60°	-120°	-80°	0°

Assignments of the turn residues ‘i+1’ and ‘i+2’ are made. However, since the torsion angles of the terminal residues—specifically ϕ of residue ‘i’ and ψ of residue ‘i+4’—are flexible, these are not included in the assignment.

Helices

Helices consist of stretches of hydrogen bonded residues with helical dihedrals. 310-helices are typically short, with one or more ‘i’-‘i+3’ hydrogen bonds, and these can be mischaracterized as turns (type I turns). In this case, mollob checks that all residues in the helix have helical dihedral angles.

Sheets

Sheets are first identified by finding hydrogen bonds between residues with sheet torsion angles. This process identifies most sheet residues. However, for strands on the edges of sheets, every second amino acid may not form an internal hydrogen bond.

To accurately identify sheet strands, mollob will find groups of sheet hydrogen bonds, then it will evaluate whether the residues are in a checkered pattern and whether the previous or subsequent residues have sheet backbone torsion angles. Thereafter, it will assign all residues in the group to a sheet classification, if no other classification has already been made. See [assign blocks](#) for details.

Assign Blocks

Secondary structure assignments are made based on hydrogen bonds. In some cases, such as the edge strands of sheets or short 310-helices, residues within a contiguous block are not assigned because they do not form an internal hydrogen bond. Mollob assign contiguous blocks of residues with the same secondary structure by testing the dihedral of residues within that block and filling gaps in assignment.

For example, a checkered sheet assignment (‘E E E E E’) will be assigned as a single contiguous β -strand (‘EEEEEEEEEE’) if all residues in the block have β -strand backbone dihedral angles. 310-helices are another example in which the ‘i’ and ‘i+3’ residues are hydrogen bonded, yet the ‘i+1’ and ‘i+2’ residues may not be. In this case, the gap will be filled by assigning residues ‘i’ through ‘i+3’ as 310-helix, if all four residues have helical dihedral angles.

Additionally, assigning blocks will label the minor classification of N- and C-terminal residues for certain secondary structure blocks, depending on the settings.

Note: The identification of the ‘N-term’ and ‘C-term’ minor classifications are done separately for residues and hydrogen bonds. These assignments may be different between residues and hydrogen bonds.

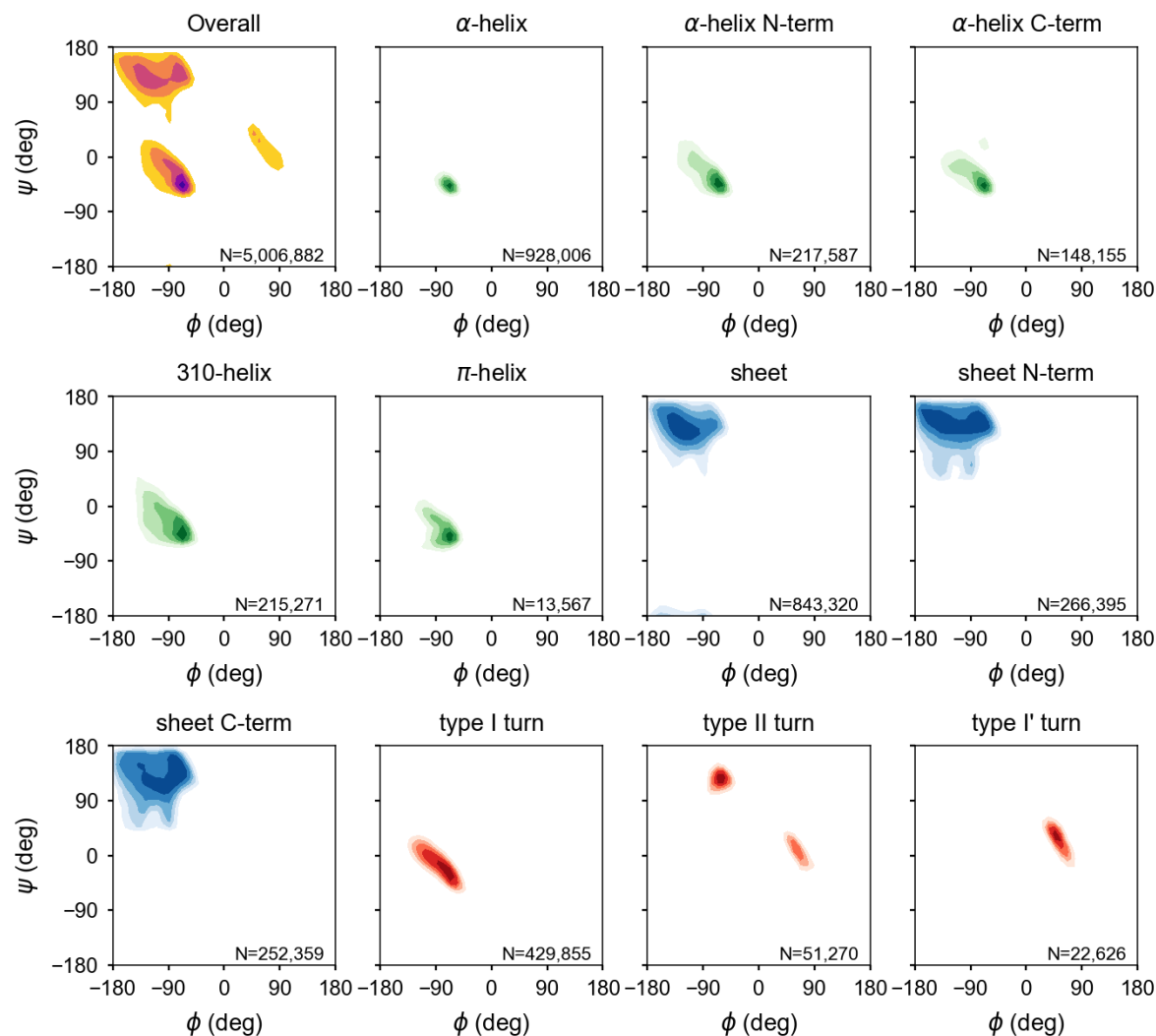
Energy Maps

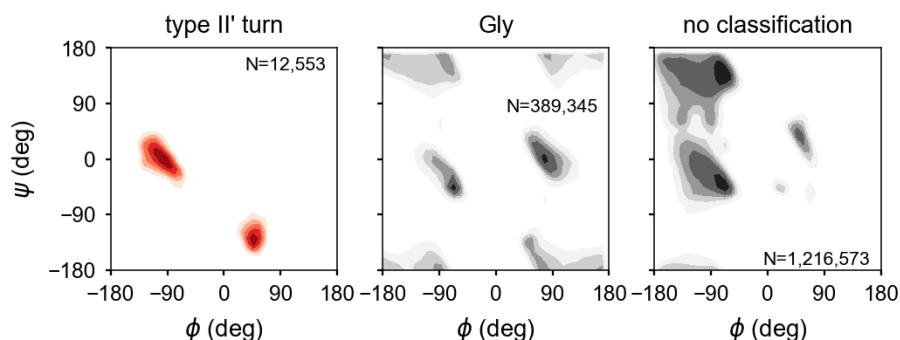
The backbone dihedral probabilities and energies are calculated from potential of mean force plots for each type of secondary structure. It is calculated from the probability of finding a particular set of dihedral angles in a group of high-resolution structures. A high probability indicates that the measured dihedral angles are observed frequently in high-resolution structures. Conversely, a low probability indicates that a particular set of dihedral angles is rarely seen in high-resolution structures. These are typically colored in yellow (relatively rare) or red (very rare).

The energies represent a potential of mean force (PMF) calculated from a Boltzmann inversion.

$$E(\Omega) = -kT \ln[P(\Omega)]$$

The energy is zero when a set of dihedrals angles is optimal for a given type of secondary structure classification. The following are the energy plots for each secondary structure classification.





hbonds command

The hbonds command detects and reports hydrogen bonds in molecules. The hbonds command has the following features:

- It identifies hydrogen bonds between backbones, sidechains, subunits and ligands.
- It has a flexible interface to identify a wide-range of hydrogen bond types including *amide*, *aliphatic*, *hydroxyl* and the specification of arbitrary electric dipole types.
- It classifies backbone-backbone hydrogen bonds based on backbone torsion angles.
- Helical stretches require contiguous residues with helical backbone torsion angles. This prevents the misclassification of 310-helices and beta turns as well as isolated i+3, i+4 and i+5 hydrogen bonds.

```
$ ml hbonds --help
usage: molliB hbonds [-h] -i id/filename [id/filename ...] [-c filename] [-l]
                    [-s] [-m [MODELS [MODELS ...]]] [--hydrogenate]
                    [--aliphatic] [--detailed] [--sort-type]

arguments:
  -h, --help            show this help message and exit
  -i id/filename [id/filename ...], --in id/filename [id/filename ...]
                        (required) The filename(s) or PDB identifier(s) of the
                        structure(s)
  -c filename, --config filename
                        The configuration filename
  -l                    List details on the molecule(s)
  -s, --save            Save fetched files to the local directory.
  -m [MODELS [MODELS ...]], --models [MODELS [MODELS ...]]
                        The models numbers to analyze.
  --hydrogenate         Strip hydrogens and re-add them before analysis

hbond options:
  --aliphatic           Includes aliphatic hydrogen bonds
  --detailed            Report detailed information on hydrogen bonds.
  --sort-type          Sort hydrogen bonds by type
```

Note: If the molecule does not have hydrogens, this command will need to be run with the `--hydrogenate` parameter. See the [hydrogenate](#) option.

Arguments

--aliphatic (Optional) Include aliphatic hydrogen bonds in the results. The acceptor-donor atom cutoff distances are elongated to 3.0 Å, and carbon atoms are allowed in hydrogen bond donor dipoles.

--detailed (Optional) Present a detailed report on the geometries of hydrogen bonds. The detailed table does not include classification information.

--sort-type (Optional) Sort the hydrogen bonds by classification type, like beta-sheet or alpha-helix.

Examples

In this example, the hemagglutinin fusion peptide structure (`-i 2KXA`) is downloaded, and its amide hydrogen bonds are reported.

```
$ ml hbonds -i 2KXA
```

Table: Hydrogen bond listing for 2KXA-1

Num	Donor	Acceptor	Classification	E (kT) / Prob.
1	A.G1.H1--N	A.G20.O--C	bb-bb amide (isolated)	1.4 / 23.6%
2	A.G1.H2--N	A.W21.O--C	bb-bb amide (isolated)	3.5 / 3.1%
3	A.A5.H--N	A.G1.O--C	bb-bb amide (isolated)	1.2 / 31.4%
4	A.I6.H--N	A.L2.O--C	bb-bb amide (alpha-helix/N-term)	3.0 / 4.8%
5	A.A7.H--N	A.F3.O--C	bb-bb amide (alpha-helix/N-term)	7.5 / 0.1%
6	A.G8.H--N	A.G4.O--C	bb-bb amide (alpha-helix)	6.2 / 0.2%
7	A.F9.H--N	A.A5.O--C	bb-bb amide (alpha-helix)	0.7 / 51.7%
8	A.I10.H--N	A.I6.O--C	bb-bb amide (alpha-helix)	1.2 / 31.5%
9	A.E11.H--N	A.A7.O--C	bb-bb amide (alpha-helix/C-term)	3.0 / 4.8%
10	A.G12.H--N	A.G8.O--C	bb-bb amide (alpha-helix/C-term)	5.6 / 0.4%
11	A.G13.H--N	A.F9.O--C	bb-bb amide (isolated)	3.2 / 4.0%
12	A.G16.H--N	A.G12.O--C	bb-bb amide (isolated)	4.6 / 1.0%
13	A.M17.H--N	A.G13.O--C	bb-bb amide (isolated)	0.0 / 100.0%
14	A.M17.H--N	A.W14.O--C	bb-bb amide (310-helix)	4.4 / 1.2%
15	A.I18.H--N	A.W14.O--C	bb-bb amide (alpha-helix/N-term)	8.9 / 0.0%
16	A.D19.H--N	A.T15.O--C	bb-bb amide (alpha-helix/N-term)	2.0 / 13.8%
17	A.G20.H--N	A.G16.O--C	bb-bb amide (alpha-helix)	0.0 / 100.0%
18	A.W21.H--N	A.M17.O--C	bb-bb amide (alpha-helix/C-term)	0.4 / 68.9%
19	A.Y22.H--N	A.I18.O--C	bb-bb amide (alpha-helix/C-term)	3.6 / 2.7%
20	A.G23.H--N	A.D19.O--C	bb-bb amide (isolated)	1.2 / 31.1%
21	A.S24.H--N	A.Y22.O--C	bb-bb amide (isolated)	5.0 / 0.7%

This example downloads a crystal structure of ubiquitin (`-i 1UBQ`), adds hydrogens to the structure and shows part (`|head -n15`) of its amide hydrogen bond report.

```
$ ml hbonds -i 1UBQ --hydrogenate|head -n15
```

Table: Hydrogen bond listing for 1UBQ

Num	Donor	Acceptor	Classification	E (kT) / Prob.
1	A.M1.H2--N	A.V17.O--C	bb-bb amide (isolated)	6.2 / 0.2%
2	A.I3.H--N	A.L15.O--C	bb-bb amide (sheet)	2.0 / 13.8%
3	A.F4.H--N	A.S65.O--C	bb-bb amide (sheet)	1.3 / 27.3%
4	A.V5.H--N	A.I13.O--C	bb-bb amide (sheet)	0.2 / 81.8%
5	A.K6.H--N	A.L67.O--C	bb-bb amide (sheet)	1.2 / 30.0%

6	A.T7.H--N	A.K11.O--C	bb-bb amide (sheet)	2.9 / 5.6%
7	A.T9.H--N	A.T7.OG1--CB	bb-sc amide (isolated)	1.0 / 36.3%
8	A.G10.H--N	A.T7.O--C	bb-bb amide (type I turn)	3.3 / 3.7%
9	A.I13.H--N	A.V5.O--C	bb-bb amide (sheet)	2.8 / 5.8%
10	A.L15.H--N	A.I3.O--C	bb-bb amide (sheet)	2.1 / 12.4%
11	A.V17.H--N	A.M1.O--C	bb-bb amide (isolated)	1.8 / 16.7%

Theory

Hydrogen bonds are defined between a donor dipole, which includes a hydrogen atom, and an acceptor dipole. The acceptor and donor dipoles are more than simple electric dipoles. The electric dipole-dipole energy is minimized when the two dipoles are aligned colinearly (head-to-toe) with respect to each other. Hydrogen bonds involve the sharing of a hydrogen atom between a donor atom (N) and an acceptor atom (O) and the distribution of electron density between these atoms and the hydrogen atom. Consequently, hydrogen bonds are arranged according to the molecular orbitals between the donor and acceptor atoms, rather than simply the electric potential between atoms. As a result, optimal hydrogen bonds are askew from a standard electric dipole.

Geometry

A hydrogen bond has an acceptor dipole and a donor dipole. The donor dipole has a d1 donor atom (H) and a d2 donor atom (N, O or C), and the acceptor has an a1 acceptor atom (O) and an a2 acceptor atom (C). The geometry of a hydrogen bond is defined by the distance of the d1a1 vector (d_{d1a1}), the θ angle and the ϕ angle.

d_{d1a1} distance (in Angstroms) The distance between the d1 donor atom (H) and the a1 acceptor atom (O).

θ angle (in degrees) The angle between the d1a1 vector and the z-axis of the acceptor dipole. The z-axis of the acceptor dipole is defined by the normalized vector formed between the acceptor a1 and a2 atoms.

ϕ angle (in degrees) The angle between the d1a1 vector and the x-axis of the acceptor dipole plane. The acceptor dipole plane is defined by the the acceptor dipole atoms and the next heaviest atom bound to atom a2.

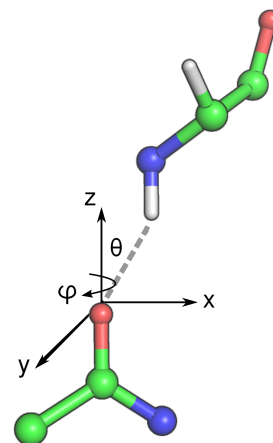


Fig. 1.1: The geometry of a hydrogen bond.

Energy and Probabilities

The reported probabilities are determined from a histogram of the d_{d1a1} distance, the θ angle and the ϕ angle compared to a set of *ca.* 11,300 high-resolution crystal structures in the PDB. A high probability indicates that the measured hydrogen bond geometries are observed frequently in high-resolution structures. Conversely, a low probability indicates that a particular hydrogen bond geometry is rarely seen in high-resolution structures. These are typically colored in yellow (relatively rare) or red (very rare).

The energies represent a potential of mean force (PMF) calculated from a Boltzmann inversion.

$$E(\Omega) = -kT \ln[P(\Omega)]$$

The energy (in units of kT) is calculated from the Boltzmann constant and temperature (kT) and the probability of finding the specific configuration, Ω . The configuration, in this case, is defined by the d_{d1a1} distance, the θ angle and the ϕ angle. The energy is 0 for an ideal hydrogen bond, and it increases as the hydrogen bond deviates from ideality—according to the probability of finding a specific geometry in high-resolution structures.

pa command

The pa command is used to fit residual dipolar couplings (RDCs) and residual anisotropic chemical shifts (RACSs, sometimes known as RCSAs) from *partially aligned* samples using NMR. The output table entries are colored for warning outliers (yellow) and bad outliers (red).

```
$ ml pa --help
usage: mollib pa [-h] -i id/filename [id/filename ...] [-c filename] [-l] [-s]
                [-m [MODELS [MODELS ...]]] [--hydrogenate] -d id/filename
                [id/filename ...] [-o filename] [-p filename] [--summary]
                [--set id]
                [--exclude [interaction-type [interaction-type ...]]]
                [--project-methyls] [--methyl-scale number]
                [--fix-sign | --nofix-sign]
                [--fix-nh-scale | --nofix-nh-scale]
                [--fix-outliers | --nofix-outliers]
```

arguments:

```
-h, --help          show this help message and exit
-i id/filename [id/filename ...], --in id/filename [id/filename ...]
                    (required) The filename(s) or PDB identifier(s) of the
                    structure(s)
-c filename, --config filename
                    The configuration filename
-l                  List details on the molecule(s)
-s, --save          Save fetched files to the local directory.
-m [MODELS [MODELS ...]], --models [MODELS [MODELS ...]]
                    The models numbers to analyze.
--hydrogenate       Strip hydrogens and re-add them before analysis
-d id/filename [id/filename ...], --data id/filename [id/filename ...]
                    (required) Alignment file or identifier with RDC and
                    RACS data
-o filename, --out filename
                    The output filename for the reports of the fit data.
-p filename, --pred filename
                    The output filename for the report of the back-
                    calculated RDCs and RACSs that are not in the
                    experimental data.
--summary           Only display the fit summary
--set id            If multiple datasets are available, this option
                    specifies which dataset to use.
--exclude [interaction-type [interaction-type ...]]
                    Exclude one or more interactions of the following
                    type(s). ex: N-H or CE-HE
--project-methyls   Fit methyl RDCs by projecting their values on the
                    corresponding C-C bond, as used by Xplor-NIH
--methyl-scale number
                    The order parameter to use in scaling the methyl RDCs.
```

fixer arguments:

```
--fix-sign          Check and fix mistakes in RDC and RACS sign
--nofix-sign        Disable check in RDC and RACS sign
--fix-nh-scale       Check and rescale couplings that were scaled to the
                    N-H RDC.
--nofix-nh-scale     Disable N-H rescaling of couplings.
--fix-outliers       Fit without outliers
--nofix-outliers     Disable fitting without outliers
```

Arguments

- d / --data filename** The file(s) with the RDC and RACS alignment data. These can be in either of the following formats:
- The molLib data format. See *Partial Alignment Data File Format*.
 - NMRPipe's DC format.
 - Magnetic resonance data files (.mr) in Xplor format submitted to the PDB. This function supports the automatic fetching and caching of magnetic resonance data files.
- o / --out filename** (Optional) The filename for the output report. The output report is rendered in Markdown.
- p / --pred filename** (Optional) The filename for the back-calculated RDCs and RACS from the SVD fit. The output report is rendered in Markdown.
- summary** (Optional) Only display the fit summary.
- exclude interactions-types** (Optional) Exclude one or more interactions types. ex: `--exclude N-H CA-HA` will exclude all N-H and CA-HA RDCs.
- set id** (Optional) Use the given data set, if multiple data sets are available. This option is useful with .mr data from the PDB, which may contain multiple alignment data sets from multiple alignment media. Sets can be selected from their alignment tensor value (ex: 500, 501, etc) or from their position within the data file, starting with 0. (ex: 0 for the first dataset, 1, for the second dataset and so on.) By default, the first dataset is used.
- project-methyls** (Optional) Use the C-C bond RDC values for the methyl ^1H - ^{13}C RDCs. This is the convention followed by X-plor NIH. By default, this is disabled.
- methyl-scale number** (Optional) The scaling constant to use in fitting the methyl RDCs. This scaling may be needed if the contribution of the C3-rotational motion was not accounted for in the reported RDCs. By default, this value is 1.0.

Note: The `models` option (`-m/--models`) will load the models as multiple molecules to be fit together in the SVD rather than conduct a separate SVD for each.

Fixer Arguments

- fix-sign / --nofix-sign** (Optional) Check to see if the sign of RDCs or RACSs of the same type need to be inverted to get a better fit. This operation is useful for automatically fixing the sign of couplings when the absolute value of the $|J+D|$ - and $|J|$ -couplings are used. By default, this fixer is **on**.
- fix-outliers / --nofix-outliers** (Optional) Check to see if there are outliers for each type of interaction. A warning outlier and a bad outlier are defined by those that give an alpha-critical cutoff of 95% and 99%, respectively, using a Grubbs test. If outliers are found, these will be removed from the fit and the reported statistics. By default, this fixer is **off**.
- fix-nh-scale / --nofix-nh-scale** (Optional) Check to see if RDCs and RACSs have been scaled to match the magnitude of N-H RDCs. If they have, scale them back down to their original values. By default, this fixer is **off**.

Partial Alignment Data File Format

The file format has the following features:

1. The interaction labels for dipolar interactions refer to two atoms (ex: 14N-H) and the interaction label for CSA interactions refer to one atom.
2. For dipolar interactions, redundant residue numbers and chain identifiers are not needed. For example, '14N-H' and '14N-14H' refer to the same dipole.
3. If the chain identifier is not specified, then the subunit 'A' is assumed.
4. Relative residue numbers are allowed. For example, '14N-C-1' is the same as the '14N-13C' dipole.
5. Errors are optional. If the error is not specified, a default value from the settings is used.

The partial alignment RDC and RACS data file has the following format:

```
# Interaction  Value (Hz)  Error (optional)
14N-H          -14.5      0.1
15N-H           3.5
A.16N-H        -8.5      0.2 # larger error

A.16H-A.15C     0.5      0.1
B.16H-B.15C     0.5      0.1

# Residual anisotropic chemical shift data
# Interaction  Value (ppb)  Error (optional)
5C              112      1
6C             -250
```

Examples

The following example fits the deposited RDCs for the hemagglutinin fusion peptide structure (-a 2KXA) to the deposited NMR structure (-i 2KXA). The output table entries are colored for warning outliers (yellow) and bad outliers (red).

```
$ ml pa -i 2KXA -d 2KXA
```

Table: Summary SVD Statistics for Molecule 2KXA-1

```
-----
Overall  Q (%): 18.2    RMS: 2.0    count: 58
N-H      Q (%): 9.3     RMS: 0.68   count: 21    Da (Hz): -8.0 Rh: 0.083
CA-CB    Q (%): 27.1    RMS: 0.48   count: 2     Da (Hz): 1.4 Rh: 0.083
CA-HA    Q (%): 16.7    RMS: 2.51   count: 26    Da (Hz): 16.4 Rh: 0.083
CB-CG    Q (%): 3.6      RMS: -      count: 1
CD-CG    Q (%): 43.4    RMS: 0.75   count: 3
HA-HA    Q (%): 30.1    RMS: 5.34   count: 3
NE-HE    Q (%): 12.8    RMS: 1.55   count: 2
Alignment Aa: -0.0003686 Ar: -3.053e-05
Saupe    Szz: -0.0007373 Syy: 0.0003228 Sxx: 0.0004144
Angles   Z (deg): 230.1 Y' (deg): 83.5 Z'' (deg): 104.9
-----
```

Table: Observed and Predicted RDCs and RACS for Molecule 2KXA-1

Interaction	Value	Error	Predicted	Deviation
A.5CA-CB	-1.8	-	-1.5	-0.3
A.7CA-CB	1.9	-	1.5	0.4
A.1CA-HA2	-20.0	-	-13.5	-6.5
A.1CA-HA3	-4.0	-	-3.1	-0.9

A.2CA-HA	-1.8	-	-2.2	0.4
A.3CA-HA	29.8	-	24.8	5.0
A.4CA-HA#	5.0	-	3.0	2.0
A.5CA-HA	-17.1	-	-14.7	-2.4
A.6CA-HA	13.3	-	15.0	-1.7
A.7CA-HA	13.2	-	15.0	-1.8
A.8CA-HA#	-11.3	-	-9.2	-2.1
A.9CA-HA	-16.8	-	-14.3	-2.5
A.10CA-HA	30.6	-	27.8	2.8
A.11CA-HA	0.6	-	-0.9	1.5
A.12CA-HA2	-12.0	-	-13.6	1.6
A.12CA-HA3	30.0	-	26.2	3.8
A.13CA-HA2	-1.0	-	0.0	-1.0
A.13CA-HA3	0.0	-	1.1	-1.1
A.14CA-HA	-16.3	-	-17.1	0.8
A.15CA-HA	-6.6	-	-10.9	4.3
A.16CA-HA#	15.1	-	14.4	0.7
A.17CA-HA	-9.8	-	-13.0	3.2
A.18CA-HA	-17.9	-	-18.0	0.1
A.19CA-HA	9.4	-	8.9	0.5
A.20CA-HA#	23.7	-	23.0	0.7
A.21CA-HA	-13.9	-	-15.1	1.2
A.22CA-HA	-11.6	-	-11.7	0.1
A.23CA-HA#	1.7	-	2.0	-0.3
A.18CB-CG2	1.2	-	1.2	0.0
A.6CD1-CG1!	-0.7	-	-1.3	0.6
A.10CD1-CG1!	-0.2	-	-1.0	0.8
A.18CD1-CG1!	-0.7	-	-1.1	0.4
A.1HA2-HA3!	-10.0	-	-14.6	4.6
A.12HA2-HA3!	15.0	-	10.8	4.2
A.13HA2-HA3!	-11.0	-	-15.2	4.2
A.3N-H	-10.5	-	-10.4	-0.1
A.4N-H	-6.8	-	-6.5	-0.3
A.5N-H	3.2	-	3.5	-0.3
A.6N-H	-3.5	-	-3.1	-0.4
A.7N-H	-7.8	-	-7.4	-0.4
A.8N-H	0.4	-	1.2	-0.8
A.9N-H	5.7	-	4.8	0.9
A.10N-H	-4.0	-	-2.6	-1.4
A.11N-H	-4.5	-	-3.2	-1.3
A.12N-H	4.4	-	4.2	0.2
A.13N-H	6.2	-	6.8	-0.6
A.14N-H	-9.9	-	-10.6	0.7
A.15N-H	-5.7	-	-6.1	0.4
A.16N-H	-13.7	-	-14.6	0.9
A.17N-H	-10.8	-	-11.1	0.3
A.18N-H	-5.1	-	-5.4	0.3
A.19N-H	-7.9	-	-8.9	1.0
A.20N-H	-14.1	-	-14.5	0.4
A.21N-H	-6.0	-	-5.6	-0.4
A.22N-H	-3.8	-	-3.8	0.0
A.23N-H	-14.1	-	-14.7	0.6
A.14NE1-HE1	-5.7	-	-7.2	1.5
A.21NE1-HE1	2.2	-	2.0	0.2

* Inverting the sign of 'N-H' interactions improved the overall Q-factor from 228.7% to 29.4%.

* Inverting the sign of 'NE-HE' interactions improved the overall Q-factor from 29.4% to 18.2%.

The following example fits the deposited RDCs for the first alignment (`--set 0`) dataset of ubiquitin (`-a 2MJB`) to the deposited NMR structure (`-i 2MJB`). The RDCs for methyl groups are projected onto the corresponding C-C bonds (`--project-methyls`) and outliers are removed from the fit (`--fix-outliers`).

```
$ ml -s pa -i 2MJB -d 2MJB --set 0 --fix-outliers --project-methyls --summary
```

Table: Summary SVD Statistics for Molecule 2MJB-1

Overall	Q (%): 23.8	RMS: 3.68	count: 477		
N-H	Q (%): 6.3	RMS: 0.52	count: 63	Da (Hz): 9.1	Rh: 0.144
C-CA	Q (%): 19.4	RMS: 0.28	count: 58	Da (Hz): -1.6	Rh: 0.144
C-H+1	Q (%): 13.0	RMS: 0.37	count: 61		
C-N+1	Q (%): 22.1	RMS: 0.19	count: 60	Da (Hz): 0.9	Rh: 0.144
CA-CB	Q (%): 15.0	RMS: 0.4	count: 38	Da (Hz): -1.6	Rh: 0.144
CA-HA	Q (%): 12.9	RMS: 2.21	count: 66	Da (Hz): -18.8	Rh: 0.144
CB-CG	Q (%): 16.8	RMS: 2.93	count: 19		
CB-HB	Q (%): 18.1	RMS: 3.34	count: 50		
CD-CG	Q (%): 31.2	RMS: 5.44	count: 19		
CD-HD	Q (%): 49.5	RMS: 9.53	count: 10		
CE-HE	Q (%): 115.0	RMS: 23.44	count: 5		
CE-SD	Q (%): 74.1	RMS: -	count: 1		
CG-HG	Q (%): 43.1	RMS: 8.04	count: 27		
Alignment	Aa: 0.0004225	Ar: 6.07e-05			
Saupe	Szz: 0.0008451	Syy: -0.0003315	Sxx: -0.0005136		
Angles	Z (deg): 49.0	Y' (deg): 22.2	Z'' (deg): 104.8		

* Inverting the sign of 'N-H' interactions improved the overall Q-factor from 351.1% to 84.6%.
 * Inverting the sign of 'C-N+1' interactions improved the overall Q-factor from 84.6% to 27.1%.
 * Removing outlier data points A.46CA-CB, A.13CB-CG2, A.16CB-HB#, A.60CB-HB#, A.48CD-HD#, A.24C-25N improved the overall Q-factor from 27.1% to 23.8%.

This example is the same as the last one, however 'CE-HE', 'CD-HD' and 'CE-SD' RDCs are excluded (`--exclude`) from the fit.

```
$ ml -s pa -i 2MJB -d 2MJB --set 0 --exclude CE-HE CD-HD CE-SD --fix-outliers \
--project-methyls --summary
```

Table: Summary SVD Statistics for Molecule 2MJB-1

Overall	Q (%): 18.7	RMS: 2.71	count: 461		
N-H	Q (%): 5.4	RMS: 0.46	count: 63	Da (Hz): 9.3	Rh: 0.147
C-CA	Q (%): 17.2	RMS: 0.25	count: 58	Da (Hz): -1.6	Rh: 0.147
C-H+1	Q (%): 13.8	RMS: 0.4	count: 61		
C-N+1	Q (%): 19.6	RMS: 0.17	count: 60	Da (Hz): 1.0	Rh: 0.147
CA-CB	Q (%): 14.4	RMS: 0.43	count: 38	Da (Hz): -1.6	Rh: 0.147
CA-HA	Q (%): 10.9	RMS: 1.9	count: 66	Da (Hz): -19.3	Rh: 0.147
CB-CG	Q (%): 16.1	RMS: 2.87	count: 19		
CB-HB	Q (%): 18.3	RMS: 3.45	count: 50		
CD-CG	Q (%): 31.8	RMS: 5.67	count: 19		
CG-HG	Q (%): 43.5	RMS: 8.29	count: 27		
Alignment	Aa: 0.0004318	Ar: 6.355e-05			
Saupe	Szz: 0.0008637	Syy: -0.0003365	Sxx: -0.0005272		
Angles	Z (deg): 48.1	Y' (deg): 21.8	Z'' (deg): 104.0		

* Inverting the sign of 'N-H' interactions improved the overall Q-factor from 367.2% to 83.2%.

* Inverting the sign of 'C-N+1' interactions improved the overall Q-factor from 83.2% to 21.1%.
 * Removing outlier data points A.46CA-CB, A.13CB-CG2, A.16CB-HB#, A.60CB-HB#, A.24C-25N improved the overall Q-factor from 21.1% to 18.7%.

Likewise, the crystal structure of ubiquitin (-i 1UBQ) can be used in the fit. In this case, the structure is missing hydrogen atoms, and these must be added (--hydrogenate).

```
$ ml -s pa -i 1UBQ -d 2MJB --set 0 --fix-outliers --project-methyls \
--hydrogenate --summary
```

Table: Summary SVD Statistics for Molecule 1UBQ

Overall	Q (%): 39.8	RMS: 5.77	count: 474		
N-H	Q (%): 14.7	RMS: 1.2	count: 62	Da (Hz): 8.9	Rh: 0.181
C-CA	Q (%): 26.2	RMS: 0.37	count: 58	Da (Hz): -1.5	Rh: 0.181
C-H+1	Q (%): 20.5	RMS: 0.56	count: 60		
C-N+1	Q (%): 31.1	RMS: 0.26	count: 61	Da (Hz): 0.9	Rh: 0.181
CA-CB	Q (%): 25.1	RMS: 0.36	count: 37	Da (Hz): -1.5	Rh: 0.181
CA-HA	Q (%): 25.8	RMS: 4.32	count: 66	Da (Hz): -18.4	Rh: 0.181
CB-CG	Q (%): 24.4	RMS: 4.18	count: 18		
CB-HB	Q (%): 63.8	RMS: 10.42	count: 51		
CD-CG	Q (%): 55.1	RMS: 9.42	count: 19		
CD-HD	Q (%): 49.6	RMS: 8.45	count: 10		
CE-HE	Q (%): 19.7	RMS: 3.68	count: 4		
CE-SD	Q (%): 36.7	RMS: -	count: 1		
CG-HG	Q (%): 94.9	RMS: 15.63	count: 27		
Alignment	Aa: 0.000412	Ar: 7.466e-05			
Saupe	Szz: 0.0008239	Syy: -0.0003	Sxx: -0.0005239		
Angles	Z (deg): 330.1	Y' (deg): 70.4	Z'' (deg): 257.7		

* Inverting the sign of 'N-H' interactions improved the overall Q-factor from 516.4% to 93.7%.
 * Inverting the sign of 'C-N+1' interactions improved the overall Q-factor from 93.7% to 46.3%.
 * Removing outlier data points A.46CA-CB, A.14CB-CG2, A.48CD-HD#, A.33CE-HE#, A.48N-H, A.7C-8H, A.28CA-CB, A.13CB-CG2, A.14CB-HB improved the overall Q-factor from 46.3% to 39.8%.

Tensor Conventions

In the absence of motion, dipolar tensors are axially symmetric (i.e. $\delta_{xx} = \delta_{yy}$) and the principal component (δ_{zz}) is colinear with the internuclear vector in the principal axis system (PAS).

Chemical shift tensors (CSA) may be axially asymmetric (i.e. $\delta_{xx} \neq \delta_{yy}$), and their geometries must be specified in relation to internal atomic coordinates. We use the convention from Cornilescu *et al.* [Cornilescu2000].

The literature reports both the chemical *shielding* tensor (σ) and the chemical *shift* tensor (δ). The difference between the two is an inversion of sign (i.e. $\sigma = -\delta$). As a result, the ordering of components between different conventions will change. In the Haeblerlen convention, the chemical shift components are ordered by their magnitudes.

$$|\delta_{zz}| \geq |\delta_{xx}| \geq |\delta_{yy}|$$

The isotropic component (δ_{iso}) has already been subtracted from the three components.

$$\delta_{iso} = \frac{1}{3} (\delta_{zz} + \delta_{xx} + \delta_{yy})$$

In the IUPAC convention, the components are normally ordered starting from the largest component (with sign) as the '11' component. However, for chemical *shielding* tensors, the '33' component is largest.

$$\sigma_{33} \geq \sigma_{22} \geq \sigma_{11}$$

MolLib uses the Haeberlen convention and chemical *shift* tensors. The backbone H, C' and N CSA tensors are defined as follows:

The $^{13}\text{C}'$ tensor (blue) has the largest component (δ_{zz}) oriented orthogonal to the O-C-N plane, and it is rotated about this component by the α_z angle.

The ^{15}N tensor (red) has the largest component (δ_{zz}) nearly colinear with the H-N bond, and it is rotated away from the bond about the yy-component (orthogonal to the H-N-C' plane) with an angle β_y .

The ^1H tensor (green) has the largest component (δ_{zz}) nearly colinear with the H-N bond, and it is rotated about the xx-component (orthogonal to the H-N-C' plane) by an angle γ_x .

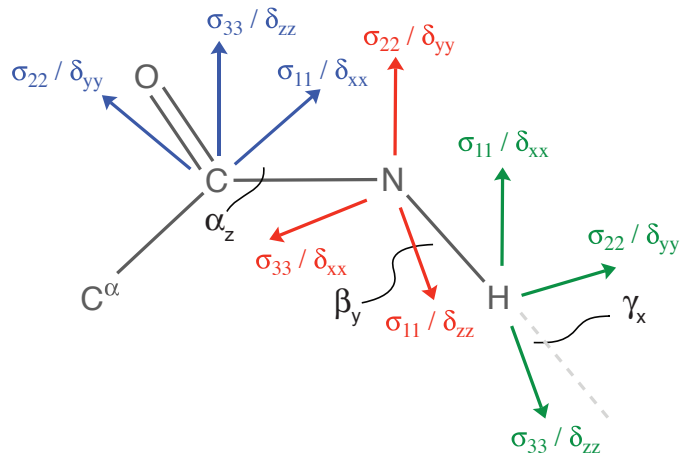


Fig. 1.2: Backbone CSA tensor conventions

Configuration Options

The following are the general configuration file settings.

Listing 1.2: [settings]

```
#: The default pH of new molecules
default_pH = 7.0

#: The default pKs of ionizable amino-acids. [Ref]_
#:
#: Some amino-acids have degenerate ionizeable atoms; these are listed and
#: separated by '-' characters. The different pKs for each ionization is
#: listed in the subsequent items in the tuple.
#:
#: .. [Ref] G. R. Grimsley, J. M. Scholtz, C. N. Pace, Protein Sci. 18, 247-51
#: (2009).
pKs = {'ASP': {'OD1-OD2': (-1.0, 3.5)},
      'GLU': {'OE1-OE2': (-1.0, 4.2)},
      'HIS': {'ND1-NE2': (6.6, 14.0)},
      'CYS': {'SG': (6.8,)},
      'TYR': {'OH': (10.3,)},
      'LYS': {'NZ': (10.5, 14.0, 14.0)},
      'last': {'O-OXT': (-1.0, 3.3)},
      'first': {'N': (7.7, 14.0, 14.0)},
      }

#: Path for the datasets
dataset_path = 'data/'
```



```

#: Path for model input molecule identifiers
model_molecule_identifiers = ('high_res.txt',
                                #'high_res_short.txt',
                                )

#: Path for the Ramachandran statistics datasets
ramachandran_dataset_path = 'data/ramachandranstatistics/'

#: Path for the Hbond statistics datasets
hbond_dataset_path = 'data/hbondstatistics/'

#: The cutoff Energy (in kT) to report good, warning and bad energies.
energy_cutoff_good = 3.4 # Within 96.6% of observed values
energy_cutoff_warning = 5.4 # Within 99.5% of observed values
energy_cutoff_bad = 20.0 # Only observed < 0.5% of the time.

#: urls to download PDB files
pdb_urls = ('https://files.rcsb.org/download/',)

#: Only load the first model, by default
pdb_first_model = True

```

The `utils.settings` impact how general functions behave and how the data are presented.

Listing 1.3: [utils.settings]

```

#: The default max width (in characters) of markdown tables (MDTable)
#: This is the value used when outputting tables outside of the terminal
default_max_width = 100

#: When writing files, overwrite them if they already exist.
overwrite_files = True

#: Save files fetched and cached files (get_or_fetch) to the local directory
save_fetched_files_locally = False

#: If files are not overwritten, the create new versions up to the following
#: number
max_file_versions = 99

```

The `hydrogens.settings` impact how hydrogen atoms are added to a molecule. These impact the `--hydrogenate` option.

Listing 1.4: [hydrogens.settings]

```

#: The default optimal length of standard bonds (in Angstroms) for biomolecules
bond_length = {'N-H': 1.023, # L. Yao, et al. JACS 130, 16518-20 (2008)
               'C-H': 1.117, # M. Ottiger, et al. JACS 121, 4690-4695 (1999)
               'C-C': 1.517, # M. Ottiger, et al. JACS 121, 4690-4695 (1999)
               'N-H2': 1.01, # PK Sawinski, et al. Crys Growth & Design
                           # 13, 1730 (2013)
               'O-H': 0.958, # MN Frey et al., JCP 59, 915-924 (1973).
               'C-N': 1.488, # MN Frey et al., JCP 59, 915-924 (1973).
               'C=O': 1.250,
               'C-O': 1.413,

```

```
}

```

The `hbonds.settings` impact how hydrogen bonds are detected. These impact the `hbonds` command and the `--rama` option in the `measure` command.

Listing 1.5: [hbonds.settings]

```
# Hydrogen Bond Definitions
# -----

#: Element selector string for the hydrogen bond donor atom1. The OR
#: operator, '|', is supported.
#:
#: Hydrogen bond dipoles are defined as:
#:
#:     donor2--donor1 .... acceptor1--acceptor2
donor1_elements = 'H|D'

#: Element selector string for the hydrogen bond donor atom2. The OR
#: operator, '|', is supported.
donor2_elements = 'N|15N|O'

#: Element selector string for the hydrogen bond acceptor atom1. The OR
#: operator, '|', is supported.
acceptor1_elements = 'O'

#: Element selector string for the hydrogen bond acceptor atom2. The OR
#: operator, '|', is supported.
acceptor2_elements = 'C|13C'

#: The cutoff distance ranges (in A) between atoms to be considered a hydrogen
#: bond.
#:
#: 'd1a1': The distance between the donor1 and acceptor1 atoms
hbond_distance_cutoff = {'d1a1': (1.5, 2.5),
                        }

#: The cutoff angle ranges (in degrees) between atoms to be considered a
#: hydrogen bond.
hbond_angle_cutoff = {'theta': (105., 180.),
                     'phi': (-180., 180.)
                     }

# Dihedral Angle Ranges
# -----

# Phi torsion angle range (in degrees) for helices. (Generously allowed)
helix_phi = (-170., 0.)

# Phi torsion angle range (in degrees) for helices. (Generously allowed)
helix_psi = (-100., 55.)

# Phi torsion angle range (in degrees) for beta-sheet. (Generously allowed)
beta_phi = (-200., -25.)

```

```

# Psi torsion angle range (in degrees) for beta-sheet. (Generously allowed)
beta_psi = (45., 220.)

#: Phi torsion angle ranges (in degrees) for residue i+1 in turns
beta_turn_i1_phi = {'turnI':  (-110., -10.),
                    'turnII': (-110., -10.),
                    'turnIp':  (10., 110.),
                    'turnIIP': (10., 110.),
                    }

#: Psi torsion angle ranges (in degrees) for residue i+1 in turns
beta_turn_i1_psi = {'turnI':  (-80., 20.),
                    'turnII': (70., 170.),
                    'turnIp':  (-20., 80.),
                    'turnIIP': (-170., -70.)
                    }

#: Phi torsion angle ranges (in degrees) for residue i+2 in turns
beta_turn_i2_phi = {'turnI':  (-140., -40.),
                    'turnII': (30., 140.),
                    'turnIp':  (40., 140.),
                    'turnIIP': (-130., -30.)
                    }

#: Psi torsion angle ranges (in degrees) for residue i+2 in turns
beta_turn_i2_psi = {'turnI':  (-50., 50.),
                    'turnII': (-50., 50.),
                    'turnIp':  (-50., 50.),
                    'turnIIP': (-50., 50.)
                    }

# Classification names
# -----

#: Classification type name for backbone-backbone (bb-bb) amide hydrogen
#: bonds
type_bb_bb_amide = 'bb-bb amide'

#: Classification type name for backbone-sidechain (sc-bb) amide hydrogen
#: bonds
type_bb_sc_amide = 'bb-sc amide'

#: Classification type name for sidechain-backbone (sc-bb) amide hydrogen
#: bonds
type_sc_bb_amide = 'sc-bb amide'

#: Classification type name for sidechain-sidechain (sc-sc) amide hydrogen
#: bonds
type_sc_sc_amide = 'sc-sc amide'

#: Classification type name for backbone-backbone (bb-bb) aliphatic hydrogen
#: bonds
type_bb_bb_aliphatic = 'bb-bb aliph.'

#: Classification type name for backbone-sidechain (bb-sc) aliphatic hydrogen

```

```

#: bonds
type_bb_sc_aliphatic = 'bb-sc aliph.'

#: Classification type name for sidechain-backbone (sc-bb) aliphatic hydrogen
#: bonds
type_sc_bb_aliphatic = 'sc-bb aliph.'

#: Classification type name for backbone-backbone (sc-sc) aliphatic hydrogen
#: bonds
type_sc_sc_aliphatic = 'sc-sc aliph.'

#: Classification type name for backbone-backbone (bb-sc) hydroxyl hydrogen
#: bonds
type_bb_sc_hydroxyl = 'bb-sc hydroxyl'

#: Classification type name for sidechain-backbone (sc-bb) hydroxyl hydrogen
#: bonds
type_sc_bb_hydroxyl = 'sc-bb hydroxyl'

#: Classification type name for sidechain-sidechain (sc-sc) hydroxyl hydrogen
#: bonds
type_sc_sc_hydroxyl = 'sc-sc hydroxyl'

#: Major classification name for type I turns
major_beta_turnI = "type I turn"

#: Major classification name for type II turns
major_beta_turnII = "type II turn"

#: Major classification name for type I' turns
major_beta_turnIp = "type I' turn"

#: Major classification name for type II' turns
major_beta_turnIIp = "type II' turn"

#: Major classification name for sheets
major_beta = 'sheet'

#: Major classification name for anti-parallel beta-sheets
major_beta_anti = 'sheet, anti-parallel'

#: Major classification name for parallel beta-sheets
major_beta_par = 'sheet, parallel'

#: Major classification name for 310-helices
major_310 = '310-helix'

#: Major classification name for alpha-helices
major_alpha = 'alpha-helix'

#: Major classification name for pi-helices
major_pi = 'pi-helix'

#: Major classification name for isolated hydrogen bonds
major_isolated = 'isolated'

#: Minor classification name for the N-terminal residues

```

```

minor_N = 'N-term'

#: Minor classification name for the C-terminal residues of alpha-helices
minor_C = 'C-term'

#: Minor classification name for glycines
minor_gly = 'Gly'

# Assign Blocks
# -----

#: Assign blocks of residues, including gaps
assign_blocks = True

#: Assign alpha-helix blocks
assign_blocks_alpha = True

#: Assign beta-strand blocks, including gaps
assign_blocks_beta = True

#: Assign 310-helix blocks, including gaps
assign_blocks_310 = True

#: assign_blocks: Overwrite classification assignments, if an assignment has
#: already been made
assign_blocks_overwrite = False

#: assign_blocks: Check the previous and subsequent residues in a contiguous
#: alpha-helix assignment
assign_blocks_alpha_extend termini = False

#: assign_blocks: Label the 'minor' classification of the specified number of
#: N- or C-terminal residues as 'N-term' or 'C-term'
assign_blocks_alpha_label_N_term = 1
assign_blocks_alpha_label_C_term = 1

#: assign_blocks: Only fill gaps in alpha-helices that are as large as the
#: following number
assign_blocks_alpha_gap_tolerance = 1

#: assign_blocks: Check the previous and subsequent residues in a contiguous
#: beta-sheet assignment
assign_blocks_beta_extend termini = True

#: assign_blocks: Label the 'minor' classification of the specified number of
#: N- or C-terminal residues as 'N-term' or 'C-term'
assign_blocks_beta_label_N_term = 1
assign_blocks_beta_label_C_term = 1

#: assign_blocks: Only fill gaps in beta-sheets that are as large as the
#: following number
assign_blocks_beta_gap_tolerance = 2

```

```

#: assign_blocks: Check the previous and subsequent residues in a contiguous
#: 310 assignment
assign_blocks_310_extend termini = False

#: assign_blocks: Label the 'minor' classification of the specified number of
#: N- or C-terminal residues as 'N-term' or 'C-term'
assign_blocks_310_label_N_term = 0
assign_blocks_310_label_C_term = 0

#: assign_blocks: Only fill gaps in 310-helices that are as large as the
#: following number
assign_blocks_310_gap_tolerance = 3

# Tables
# -----

#: Render HBond tables with detailed information
hbond_table_detailed = False

#: Sort HBond tables by the hydrogen bond type
hbond_table_sort_type = False

#: Render the Rama table with detailed information, including minor assignments
rama_table_detailed = True

```

The `pa.settings` impact how partial alignment (RDC and RACS) data are fit.

Listing 1.6: `[pa.settings]`

```

#: The gyromagnetic ratios (in rad T^-1 s^-1) of common nuclei
gamma = {'H': 267.513E6,
         'C': 67.262E6,
         'N': -27.116E6}

#: Calculate dipolar couplings from bond lengths and gyromagnetic ratios. If
#: this is False, the values in the default_predicted_rdc are used. However,
#: if a value is not specified in the default_predicted_rdc, the value will
#: still be calculated from bond lengths.
calculate_from_bonds = False

#: The default dipolar couplings calculated.
#: The static dipolar coupling constants listed are for common bonds in
#: proteins. Values are calculated based on average calculated DCCs for 2MJB
#: and scaled to an HN DCC of -11472Hz. This is equal to the
#: dipolar reduced anisotropy.
#: These values are used if ``calculate_from_bonds`` is False.
default_predicted_rdc = {'N-H': 10823., #11472., # 1.02 A
                        'NE1-HE1': 10823.,
                        'CA-HA': -22300.,
                        'C-N+1': 1115.,
                        'C-CA': -1880.,
                        'CA-CB': -1880.,
                        }

#: The default RACS tensor values of various nuclei in proteins.

```

```

#: order - the xx/yy/zz component order of the tensor.
#:
#:      1. The first component is colinear with the atom -- ref_atom1
#:          vector
#:
#:      2. The second component is orthogonal to the
#:          atom -- ref_atom1 -- ref_atom2 plane
#:
#:      3. The third component is orthogonal to the second two components.
#:
#:      4. Tensor rotations conducted with the Z-Y-X convention for the
#:          alpha, beta and gamma angles, respectively.
#:
#: See the following reference for tensor conventions and values: Cornilescu
#: et al. JACS 2000, 122, 10143.

default_predicted_racs = {
    # The Carbonyl CSA. This convention places the x-axis (s11) near the,
    # C-N bond and the z-axis (s33) orthogonal to the O-C-N plane.
    'C': {'delta': -86.53 * 1.03, # ppm (Reduced anisotropy)
          'eta': 0.63,
          'alpha': 40.,          # degrees
          'beta': 0.,           # degrees
          'gamma': 0.,          # degrees
          'ref_atom1': 'N+1',
          'ref_atom2': 'O',
          'order': 'xzy',
        },

    # The Nitrogen CSA. This convention places the z-axis (s11) near the H-N
    # bond, and the y-axis (s22) orthogonal to the H-N-CA plane.
    'N': {'delta': 108.53 * 0.993, # ppm (Reduced anisotropy)
          'eta': 0.16,
          'alpha': 0.,            # degrees
          'beta': -20.,          # degrees
          'gamma': 0.,           # degrees
          'ref_atom1': 'H',
          'ref_atom2': 'CA',
          'order': 'zyx',
        },

    # The amide H CSA. This convention places the z-axis (s33) near the H-N
    # bond, and the x-axis (s11) orthogonal to the H-N-C plane.
    'H': {'delta': -5.93,        # ppm (Reduced anisotropy)
          'eta': 1.00,
          'alpha': 0.,           # degrees
          'beta': 0.,           # degrees
          'gamma': -7.,          # degrees
          'ref_atom1': 'N',
          'ref_atom2': 'C-1',
          'order': 'xzy',
        },
    }

#: Default errors in absolute values
default_error = {'N-H': 0.2, # Hz
                 'CA-HA': 0.5, # Hz

```

```

        'N-C': 0.5, # Hz
        'CA-C': 0.5, # Hz
        'NE-HE': 0.2, # Hz
        'C-C': 0.5, # Hz (bond type)
        'C-H': 1.0, # Hz (bond type)
        'C': 1., # ppb
        'N': 1., # ppb
        'H': 5., # ppb
    }

#: urls to download mr files
mr_urls = ('https://files.rcsb.org/download/',)

#: Xplor-NIH incorporates C-H methyl RDCs by projecting them on the associated
#: carbon-carbon bond. If the data uses this convention, then set this setting
#: to True. This convention, however, occludes the use of C-C RDCs, like the
#: CA-CB coupling of an ALA, and it is not set by default.
project_methyls = False

#: The order parameter to use for methyl groups
methyl_order_parameter = 1.0

#: Settings to enable individual fixers by default
enable_signfixer = True
enable_outlierfixer = False
enable_nhscalefixer = False

#: The alpha-critical values for the Grubbs test of outliers. These are used to
#: identify warning or bad data points in terms of their deviations with
#: respect to the best-fit SVD data.
alpha_warning = 0.05
alpha_bad = 0.01

```

The following options only impact the creation of mollib datasets through the `build_data` setup command.

Listing 1.7: [statistics.settings]

```

#: The number of bins to use (along each dimension) in creating the
#: Ramachandran histogram
ramachandran_histogram_bins = 72

#: The number of bins to use (along each dimension) in creating the
#: hydrogen bond histogram
hbond_histogram_bins = 8

```


Command-line Tips and Tricks

Shell Aliases and Functions

The following aliases are helpful in reducing typing on the command line. These can be added to the user shell script (ex: `.bashrc`).

```
# mollib: backbone dihedral angles
alias mlr='mollib measure --rama -i'

# mollib: hydrogen bonds
alias mlh='mollib hbonds --hydrogenate -i'

# mollib: partial alignment from the PDB
function mlp() { mollib pa -i $1 -a $1 ${@:2};}

# mollib: download a PDB file
function mlget() { mollib process -i $1 --save; gzip -d $1.pdb.gz;}
```

Python Recipes

Loading a Molecule

A molecule can be loaded directly from molecule objects. The following will automatically download the file, cache it and read in the PDB for the structure '2kxa'.

```
>>> from mollib import Molecule
>>> mol = Molecule('2kxa')
```

This structure includes 10 models, and the first model is loaded by default. Other models can be loaded.

```
>>> from mollib import Molecule
>>> mol = Molecule('2kxa', model_id=3)
```

Loading Multiple Models

A list of models can be loaded using the `mollib.MoleculeReader` factory directly.

```
>>> from mollib import MoleculeReader
>>> mr = MoleculeReader()
>>> all_models = mr.read('2kxa')
>>> print(len(all_models))
10

>>> some_models = mr.read('2kxa', model_ids=[1, 5, 8])
>>> for model in some_models:
...     print(model)
Molecule (2kxa-1):      1 chains, 24 residues, 332 atoms.
Molecule (2kxa-5):      1 chains, 24 residues, 332 atoms.
Molecule (2kxa-8):      1 chains, 24 residues, 332 atoms.
```

Accessing Chains, Residues, Atoms and Coordinates

Molecules, chains, and residues are extended dict objects. Chains are accessed by chain id (ex: 'A'), residues are accessed by residue number (ex: 3), and atoms are accessed by atom name (ex: 'CA').

```
>>> from mollib import Molecule
>>> mol = Molecule('2kxa') # loads the first model
>>> chains = list(mol.chains)
>>> print(chains)
[A]

>>> chainA = mol['A']
>>> residues = list(chainA.residues)
>>> print('{} , ...'.format(residues[:5]))
[G1, L2, F3, G4, A5], ...

>>> F3 = chainA[3]
>>> atoms = list(F3.atoms)
>>> print('{} , ...'.format(atoms[:5]))
[A.F3.N, A.F3.CA, A.F3.C, A.F3.O, A.F3.CB], ...

>>> CA = F3['CA']
>>> print("position: {:.1f} {:.1f} {:.1f}".format(CA.x, CA.y, CA.z))
position: 13.2 -2.7 6.3

>>> mol['A'][3]['CA'] == CA
True
```

The following pages describe the changes and new features in new releases of mollib.

Mollib 1.0 release notes

Aug 13, 2017

Version 1.0 of mollib is the first release. This page documents the changes and features available in this version.

Minor Releases

- 1.0.7 - Fixed a bug in defining the topology of Ser residues.
- 1.0.6 - Fixed deployment of manylinux1 and osx releases.

Compatibility

Mollib ver 1.0 is compatible with Python 2.7 and 3.3+.

Requirements

- numpy
- scipy

CLI Features

The command-line interface (CLI) includes a series of plugins for processing and analyzing molecular structures.

process

- The *process command* plugin is used to read molecular structures and modify them.
 - Currently supports the loading of protein databank (PDB) files as well as the fetching and caching of PDB files from the internet.
 - The hydrogens plugin (`--hydrogenate`) is included to strip and replace hydrogen atoms in a molecule. The topology of protein molecules are included and the CONECT topologies in PDB files are supported.
 - Individual models or many models can be selected.
-

measure

- The *measure command* plugin is used to measure interatomic distances, angles and dihedrals in molecules.
 - Statistical summaries of measurements are available with the `--stats` option.
 - Ramachandran angles are calculated with the `--rama` option. The Ramachandran angles include secondary structure assignments based on the backbone dihedral angles and hydrogen bonding pattern, like DSSP.
 - Statistics on the frequency of the observed Ramachandran angles for the given secondary structure type are provided. Statistics are calculated from the highest resolution structures from the PDB. (Unique structures with resolutions between 0.5-1.6Å, observed R-factors better than 0.25 and free R-factors better than 0.30.)
-

hbonds

- The *hbonds* plugin is used to measure and rate amide and aliphatic hydrogen bonds in structures.
 - Hydrogen bonds are classified based on backbone torsion (Ramachandran) angles and the acceptor-donor position in the primary sequence.
 - Statistics on the frequency of observed hydrogen bonds, based on their interatomic distances and angles, are provided. Statistics are calculated from the highest resolution structures from the PDB, where hydrogens were added with the hydrogens plugin. (Unique structures with resolutions between 0.5-1.6Å, observed R-factors better than 0.25 and free R-factors better than 0.30.)
-

pa

- The *pa* (partial alignment) plugin is used to fit residual dipolar coupling (RDC) and residual chemical shift anisotropy (RACS) data of biomolecules using a Singular Value Decomposition (SVD).
- Supports the refinement of sum couplings, like those of methylenes, with the `#` wildcard operator. ex: `'CA-HA#'`
- Supports the refinement of multiple molecular structures simultaneously with one dataset.
- Includes detailed statistics on the fit for each type of interaction.
- Identifies outliers based on Grubbs tests.
- Includes 'fixers' to find and fix errors in the sign of RDCs and RACS as well as the exclusion of outliers.
- Reads data from mollib partial alignment files (`.pa`), NMRPipe DC input files and magnetic resonance data in Xplor-NIH format.

- Supports the automatic fetching of magnetic resonance data from the PDB. (i.e. .mr files) However, only residue numbers are read in at this time.
- Currently includes CSA tensor data for the backbone H, C, and N atoms of proteins. Other nuclei can be easily integrated through a settings file.
- Supports methyl RDC data that have been projected onto C-C (or S-C) bonds, as used in Xplor-NIH.
- Weights the contribution of RDCs and RACS based on relative errors. If errors are not specified, default errors are used from the settings.
- Pre-calculated dipolar couplings can be used from the settings, or dipolar couplings can be calculated from the bond lengths and gyromagnetic ratios of the spins from an interaction.
- Interactions are identified using an abbreviated syntax. For example, 14N-H, A. 14N-H, 14N-14H and 14N-C-1 are all valid identifiers.
- Alignment files with multiple alignment media datasets can be selected with the `--set` option.

API Features

The API has the following features and updates.

- *Testing Framework.* A testing framework that uses pytest, tox and a CLI tester.
- *CLI Tester.* The CLI tester is located in the `tests/cli` directory, and it is invoked by typing `make test-cli` in the root project directory. It is also included in the tox tests under the `cli` environment.
- *Plugin Framework.* A plugin framework to add and remove functionality. The base Plugin is located in the `plugins` folder.
- *Settings Manager.* A settings manager to dynamically change settings from the command line or through configuration files.
- *Optimized Functions.* Optimized (Cython) functions to calculate the length of vectors and to find atoms within a distance cutoff located in `molLib.core`.
- *Statistics Plugin.* A statistics plugin to calculate statistics and datasets over high-resolution structures.

The following are instructions for locally developing a branch of mollib.

1. **Checkout.** First, checkout the branch using git.
2. **Build Cython Extensions.** The Cython and C extensions can be compiled in place.

```
$ make inplace
```

3. **Build Datasets** Build and compute the datasets. This may take a few hours.

```
$ make build-data
```

4. **Build Documentation** Build the documentation in html format under docs/

```
$ make docs
```

Note: You will need to have sphinx, sphinxcontrib-napoleon, latex and latexmk installed.

5. **Test the Build** Run the package's tests.

```
$ make test
```

Note: You will need nosetests to run the tests.

Alternatively, mollib can be tested against multiple platforms using [tox](#).

```
$ make test-all
```

6. **Install in Developer Mode.** Install the package in developer mode. This adds the package's source path to the python path. Edits to the source path are reflected in the global script.

```
$ make develop
```

If you'd like to uninstall the develop mode, use the following command.

```
$ python setup.py develop --uninstall
```

Testing

Molib includes 4 different kinds of tests. These are all executed by the `make test-all` command.

1. **Pytests and unittests.** These are tests stored in the `tests` directory. They are run automatically by executing one of the following:
 - (a) `pytest`
 - (b) `make test`
 - (c) `make test-all`
2. **Docstring tests.** These are tests within the docstrings of functions, classes and methods in the `molib` source (`molib` subdirectory). These tests are run automatically by executing one of the following:
 - (a) `pytest`
 - (b) `make test`
 - (c) `make test-all`
3. **Tox.** All of the pytests, unittests and doctests are tested in each Python environment using `tox`. These tests are run automatically by executing one of the following:
 - (a) `tox`
 - (b) `make test-all`
4. **CLI tests.** These tests detect changes in the output text from a specific set of command arguments and `molib`. These tests are located in `tests/cli` directory. An error will be flagged if the output of a `molib` command has changed from the contents of the `.txt` file. The output of the commands are also used in the documentation within the `.rst` files. These tests are run by executing:
 - (a) `make test-cli`
 - (b) Additionally, if the output of a command changes, the reference commands can be reset by entering the `tests/cli` directory and typing `make clean&&make build`. The changed `.txt` and `.rst` files should be committed to the repository.
 - (c) New commands can be created by typing the command in a `.sh` file and making this file executable.

Makefile Options

The `make` command contains a number of commands to setup and develop `molib`. The `make` commands are listed by typing `make help`.

Including Datasets

Datasets are included in the `mollib/data` directory. Data files should be included in the `MANIFEST.in` file.

Adding Plugins

Adding plugin modules may require the registration of the plugin, the registration of the plugin's settings or both.

To register a plugin, add the following to the root `__init__.py` file for the plugin:

```
from .plugin import PluginClass
plugin = PluginClass()
```

To register a plugin's settings, add the following to the root `__init__.py` file for the plugin:

```
from . import settings

from mollib.core import register_settings
register_settings(settings)
```

Branches

Molib uses git-flow to organize branches. These are the main branches:

1. **master**. The master branch is the default branch, and it contains the *production* code. It should pass all tests including travis-ci tests.
2. **development**. The development branch is used to accept new features, hotfixes and releases. It should pass all tests including travis-ci tests.

Docstring Format

Docstrings follow the numpy style. There are a few additional guidelines:

1. dict parameters and return values should list the expected keys/values
 1. dict parameters should list the key and value types using 'key' and 'value' in bold. If known, the object type should be listed after the description.

```
- **key**: interaction label, str
```

2. dict return values should either list the key/value pairs, or list specific keys and values.

```
- 'Q (%)': The fit Q-factor in percentage, float
```

2. *Sublists* should have a new line before the sublisting.


```
- 'Overall': Overall Statistics, :obj:`collections.OrderedDict`

- 'Q (%)': The fit Q-factor in percentage, float
- 'RMS': The root-mean square of the fit (Hz/ppb), float
- 'count': The number of interactions fit, int
```

3. *Lists* that follow a paragraph listing in a parameter should not be indented with respect to the paragraph.

```
angles: dict
    A dict of the angles between atoms that define the hydrogen
    bond.

- **key**: tuple of three :obj:`Atom` objects
- **value**: the angle (in deg) between the :obj:`Atom` objects
```

4. *args* and *kwargs* args are listed separately and as optional parameters.

```
Parameters
-----
args: tuple, optional
    If specified a default argument, then this will be returned if
    the key isn't found. Otherwise a ValueError exception is raised.
kwargs: dict, optional
    If specified a default argument, then this will be returned if
    the key isn't found. Otherwise a ValueError exception is raised.
```

Example 1

```
def calc_summary(magnetic_interactions, Saupe_components, data, predicted):
    """Calculate the statistics between predicted and calculated RDCs and
    RACSS.

    Parameters
    -----
    magnetic_interactions: list of dicts
        - A list of dicts, one for each molecule to be fit.
        See :class:`mollib.pa.process_molecule.Process`
    Saupe_components: dict
        See the output of :func:`mollib.pa.svd.calc_pa_SVD`
    data: dict
        - **key**: interaction labels, str
        - **value**: :obj:`mollib.pa.RDC` or :obj:`mollib.pa.RACS` data
          values.
    predicted: dict
        - **key**: interaction labels, str
        - **value**: :obj:`mollib.pa.RDC` or :obj:`mollib.pa.RACS` data
          values.

    Returns
    -----
    summary: :obj:`collections.OrderedDict`

        - 'Overall': Overall Statistics, :obj:`collections.OrderedDict`
```

```

- 'Q (%)': The fit Q-factor in percentage, float
- 'RMS': The root-mean square of the fit (Hz/ppb), float
- 'count': The number of interactions fit, int

- 'Alignment': Details on the alignment tensor,
  :obj:`collections.OrderedDict`

  - 'Aa': The alignment tensor anisotropy, float
  - 'Ar': The alignment tensor rhobicity, float

- 'Saupe': Details on the Saupe matrix, :obj:`collections.OrderedDict`

  - 'Szz': The zz-component of the Saupe matrix, float
  - 'Sxx': The xx-component of the Saupe matrix, float
  - 'Syy': The yy-component of the Saupe matrix, float

- 'Angles': Alignment tensor orientation in Rose convention,
  :obj:`collections.OrderedDict`

  - "Z (deg)": The alignment alpha angle (deg), float
  - "Y (deg)": The alignment beta angle (deg), float
  - "Z'" (deg)": The alignment gamma angle (deg), float
"""

```

Example 2

```

def fill_gaps(molecule, classifications, classification_type, dihedral_test,
              extend_terminii=False, label_N_term=0, label_C_term=0,
              gap_tolerance=1, overwrite_assignments=False):
    """Fill gaps in the classifications dict assignments.

    Gaps occur in the secondary structure assignment from hydrogen bonds,
    for example, with beta-strands on the edges of beta sheets. This
    function finds stretches of secondary structure assignments, it checks
    the dihedral angles and fills in gaps in the stretches. For a sheet:
    'E E E E' becomes 'EEEEEEE'.

    Parameters
    -----
    molecule: :obj:`mollib.Molecule`
        The molecule object to classify the secondary structure elements.
    classifications: dict
        A dict with the classifications.

        - **key**: (chain.id, residue.number). ex: ('A', 31)
        - **value**: (major_classification, minor_classification).
          ex: ('alpha-helix', 'N-term')
    classification_type: str
        The name of the classification type to check. ex: 'alpha-helix'
    dihedral_test: function or None
        - A test function that takes a :obj:`mollib.Residue` and returns
          True if the residue's dihedral angles are within range for the
          'classification_type'.
        - If None is specified, then the dihedral angles of residues will
    """

```

```

        not be tested.
    extend_terminii: bool or int, optional
        If True, the previous and subsequence residues of each contiguous
        stretch of residue classification will be checked to see if they fall
        within the dihedral angle range as well.
    label_N_term: int (optional)
        Label the first 'N' number of residues in the contiguous block as
        'N-term'
    label_C_term: int, optional
        Label the last 'N' number of residues in the contiguous block as
        'C-term'
    gap_tolerance: int, optional
        The assignment of contiguous stretches of a secondary structure
        assignment will tolerate this number of 'gaps' in the residue
        numbers.
        For a gap_tolerance of 1 and a checked sheet assignment, the
        following group 'E E E E' will be treated as a single contiguous
        block of sheetassignments.
    overwrite_assignments: bool, optional
        If True, classification assignments will be overwritten, if an
        assignments has already been made for a given residue.

```

Returns

None

"""

Building and Deployment

Building

Platform specific distributions are needed because mollib includes C extensions written in Cython. These extension must be compiled using a compiler for each type of operating system.

The build and install distribution packages, the following packages are needed:

- [Cython 0.25+](#)

Linux Distributions (Python Wheels)

Compile and build a Python wheel package.

```
$ python setup.py bdist_wheel
```

The Python wheel can be installed using [pip](#).

```
$ sudo pip install <package_file.whl>
```

Mac OS X Distributions (mpkg)

1. Upload the source distribution

```
$ python setup.py sdist upload
```

2. Upload a binary distribution

```
$ python setup.py bdist_wheel upload
```

2. Compile and build a Mac OS X package (.mpkg) file.

```
$ python setup.py bdist_mpkg
```

The .mpkg file can be installed by dragging it into the Applications folder.

Deployment

The linux and osx packages are built using the mollib-wheels repository. Follow these steps to create a new release.

1. **mollib repository.** Tag a new release on the mollib *master* branch and bump the version in the mollib/__version__.py file.
2. **mollib-wheelhouse.** Update the BUILD_COMMIT in the .travis.yml to point to the new mollib tag. If needed, switch the twine repository to the test servers.
3. Push the sdist to PyPi.

Core Module

Atom

Residue

Chain

Molecule

Reader Factories

Base Reader

PDB Readers

Functions

Plugins

Settings

Most submodules have a `settings.py` file. These contain a listing of annotated variables and values.

The `SettingsManager` is called by the main program (`__main__`) to read in configuration files.

Note: Settings should be retrieved directly from the corresponding settings modules and not copied locally in the code. This ensures that settings read in from the configuration files are properly loaded by modules.

molib/core/settings.py

```

#: The default pH of new molecules
default_pH = 7.0

#: The default pKs of ionizable amino-acids. [Ref]_
#:
#: Some amino-acids have degenerate ionizeable atoms; these are listed and
#: separated by '-' characters. The different pKs for each ionization is
#: listed in the subsequent items in the tuple.
#:
#: .. [Ref] G. R. Grimsley, J. M. Scholtz, C. N. Pace, Protein Sci. 18, 247-51
#: (2009).
pKs = {'ASP': {'OD1-OD2': (-1.0, 3.5)},
      'GLU': {'OE1-OE2': (-1.0, 4.2)},
      'HIS': {'ND1-NE2': (6.6, 14.0)},
      'CYS': {'SG': (6.8,)},
      'TYR': {'OH': (10.3,)},
      'LYS': {'NZ': (10.5, 14.0, 14.0)},
      'last': {'O-OXT': (-1.0, 3.3)},
      'first': {'N': (7.7, 14.0, 14.0)},
      }

#: Path for the datasets
dataset_path = 'data/'

#: Path for model input molecule identifiers
model_molecule_identifiers = ('high_res.txt',
                               #'high_res_short.txt',
                               )

#: Path for the Ramachandran statistics datasets
ramachandran_dataset_path = 'data/ramachandranstatistics/'

#: Path for the Hbond statistics datasets
hbond_dataset_path = 'data/hbondstatistics/'

#: The cutoff Energy (in kT) to report good, warning and bad energies.
energy_cutoff_good = 3.4 # Within 96.6% of observed values
energy_cutoff_warning = 5.4 # Within 99.5% of observed values
energy_cutoff_bad = 20.0 # Only observed < 0.5% of the time.

#: urls to download PDB files
pdb_urls = ('https://files.rcsb.org/download/',)

#: Only load the first model, by default
pdb_first_model = True

```

Plugins Module

Classes

Hydrogens Module

Functions

Settings

mollib/hydrogens/settings.py

```
#: The default optimal length of standard bonds (in Angstroms) for biomolecules
bond_length = {'N-H': 1.023,    # L. Yao, et al. JACS 130, 16518-20 (2008)
               'C-H': 1.117,    # M. Ottiger, et al. JACS 121, 4690-4695 (1999)
               'C-C': 1.517,    # M. Ottiger, et al. JACS 121, 4690-4695 (1999)
               'N-H2': 1.01,    # PK Sawinski, et al. Crys Growth & Design
                               # 13, 1730 (2013)
               'O-H': 0.958,    # MN Frey et al., JCP 59, 915-924 (1973).
               'C-N': 1.488,    # MN Frey et al., JCP 59, 915-924 (1973).
               'C=O': 1.250,
               'C-O': 1.413,
               }
```

Hbonds Module

Find Hydrogen Bonds

Classify Hydrogen Bonds

Classify Secondary Structure

Tables

Hbond Table

Ramachandran Table

Settings

mollib/hbonds/settings.py

```
# Hydrogen Bond Definitions
# -----

#: Element selector string for the hydrogen bond donor atom1. The OR
#: operator, '|', is supported.
#:
#: Hydrogen bond dipoles are defined as:
```



```

#:
#:      donor2--donor1 .... acceptor1--acceptor2
donor1_elements = 'H|D'

#: Element selector string for the hydrogen bond donor atom2. The OR
#: operator, '|', is supported.
donor2_elements = 'N|15N|O'

#: Element selector string for the hydrogen bond acceptor atom1. The OR
#: operator, '|', is supported.
acceptor1_elements = 'O'

#: Element selector string for the hydrogen bond acceptor atom2. The OR
#: operator, '|', is supported.
acceptor2_elements = 'C|13C'

#: The cutoff distance ranges (in A) between atoms to be considered a hydrogen
#: bond.
#:
#: 'd1a1': The distance between the donor1 and acceptor1 atoms
hbond_distance_cutoff = {'d1a1': (1.5, 2.5),
                        }

#: The cutoff angle ranges (in degrees) between atoms to be considered a
#: hydrogen bond.
hbond_angle_cutoff = {'theta': (105., 180.),
                     'phi': (-180., 180.)
                     }

# Dihedral Angle Ranges
# -----

# Phi torsion angle range (in degrees) for helices. (Generously allowed)
helix_phi = (-170., 0.)

# Phi torsion angle range (in degrees) for helices. (Generously allowed)
helix_psi = (-100., 55.)

# Phi torsion angle range (in degrees) for beta-sheet. (Generously allowed)
beta_phi = (-200., -25.)

# Psi torsion angle range (in degrees) for beta-sheet. (Generously allowed)
beta_psi = (45., 220.)

#: Phi torsion angle ranges (in degrees) for residue i+1 in turns
beta_turn_i1_phi = {'turnI': (-110., -10.),
                   'turnII': (-110., -10.),
                   'turnIp': (10., 110.),
                   'turnIIp': (10., 110.),
                   }

#: Psi torsion angle ranges (in degrees) for residue i+1 in turns
beta_turn_i1_psi = {'turnI': (-80., 20.),
                   'turnII': (70., 170.),

```

```

        'turnIp': (-20., 80.),
        'turnIip': (-170., -70.)
    }

#: Phi torsion angle ranges (in degrees) for residue i+2 in turns
beta_turn_i2_phi = {'turnI': (-140., -40.),
                    'turnII': (30., 140.),
                    'turnIp': (40., 140.),
                    'turnIip': (-130., -30.)
                    }

#: Psi torsion angle ranges (in degrees) for residue i+2 in turns
beta_turn_i2_psi = {'turnI': (-50., 50.),
                    'turnII': (-50., 50.),
                    'turnIp': (-50., 50.),
                    'turnIip': (-50., 50.)
                    }

# Classification names
# -----

#: Classification type name for backbone-backbone (bb-bb) amide hydrogen
#: bonds
type_bb_bb_amide = 'bb-bb amide'

#: Classification type name for backbone-sidechain (sc-bb) amide hydrogen
#: bonds
type_bb_sc_amide = 'bb-sc amide'

#: Classification type name for sidechain-backbone (sc-bb) amide hydrogen
#: bonds
type_sc_bb_amide = 'sc-bb amide'

#: Classification type name for sidechain-sidechain (sc-sc) amide hydrogen
#: bonds
type_sc_sc_amide = 'sc-sc amide'

#: Classification type name for backbone-backbone (bb-bb) aliphatic hydrogen
#: bonds
type_bb_bb_aliphatic = 'bb-bb aliph.'

#: Classification type name for backbone-sidechain (bb-sc) aliphatic hydrogen
#: bonds
type_bb_sc_aliphatic = 'bb-sc aliph.'

#: Classification type name for sidechain-backbone (sc-bb) aliphatic hydrogen
#: bonds
type_sc_bb_aliphatic = 'sc-bb aliph.'

#: Classification type name for backbone-backbone (sc-sc) aliphatic hydrogen
#: bonds
type_sc_sc_aliphatic = 'sc-sc aliph.'

#: Classification type name for backbone-backbone (bb-sc) hydroxyl hydrogen
#: bonds
type_bb_sc_hydroxyl = 'bb-sc hydroxyl'

```

```

#: Classification type name for sidechain-backbone (sc-bb) hydroxyl hydrogen
#: bonds
type_sc_bb_hydroxyl = 'sc-bb hydroxyl'

#: Classification type name for sidechain-sidechain (sc-sc) hydroxyl hydrogen
#: bonds
type_sc_sc_hydroxyl = 'sc-sc hydroxyl'

#: Major classification name for type I turns
major_beta_turnI = "type I turn"

#: Major classification name for type II turns
major_beta_turnII = "type II turn"

#: Major classification name for type I' turns
major_beta_turnIp = "type I' turn"

#: Major classification name for type II' turns
major_beta_turnIIp = "type II' turn"

#: Major classification name for sheets
major_beta = 'sheet'

#: Major classification name for anti-parallel beta-sheets
major_beta_anti = 'sheet, anti-parallel'

#: Major classification name for parallel beta-sheets
major_beta_par = 'sheet, parallel'

#: Major classification name for 310-helices
major_310 = '310-helix'

#: Major classification name for alpha-helices
major_alpha = 'alpha-helix'

#: Major classification name for pi-helices
major_pi = 'pi-helix'

#: Major classification name for isolated hydrogen bonds
major_isolated = 'isolated'

#: Minor classification name for the N-terminal residues
minor_N = 'N-term'

#: Minor classification name for the C-terminal residues of alpha-helices
minor_C = 'C-term'

#: Minor classification name for glycines
minor_gly = 'Gly'

# Assign Blocks
# -----

#: Assign blocks of residues, including gaps
assign_blocks = True

```

```

#: Assign alpha-helix blocks
assign_blocks_alpha = True

#: Assign beta-strand blocks, including gaps
assign_blocks_beta = True

#: Assign 310-helix blocks, including gaps
assign_blocks_310 = True

#: assign_blocks: Overwrite classification assignments, if an assignment has
#: already been made
assign_blocks_overwrite = False

#: assign_blocks: Check the previous and subsequent residues in a contiguous
#: alpha-helix assignment
assign_blocks_alpha_extend termini = False

#: assign_blocks: Label the 'minor' classification of the specified number of
#: N- or C-terminal residues as 'N-term' or 'C-term'
assign_blocks_alpha_label_N_term = 1
assign_blocks_alpha_label_C_term = 1

#: assign_blocks: Only fill gaps in alpha-helices that are as large as the
#: following number
assign_blocks_alpha_gap_tolerance = 1

#: assign_blocks: Check the previous and subsequent residues in a contiguous
#: beta-sheet assignment
assign_blocks_beta_extend termini = True

#: assign_blocks: Label the 'minor' classification of the specified number of
#: N- or C-terminal residues as 'N-term' or 'C-term'
assign_blocks_beta_label_N_term = 1
assign_blocks_beta_label_C_term = 1

#: assign_blocks: Only fill gaps in beta-sheets that are as large as the
#: following number
assign_blocks_beta_gap_tolerance = 2

#: assign_blocks: Check the previous and subsequent residues in a contiguous
#: 310 assignment
assign_blocks_310_extend termini = False

#: assign_blocks: Label the 'minor' classification of the specified number of
#: N- or C-terminal residues as 'N-term' or 'C-term'
assign_blocks_310_label_N_term = 0
assign_blocks_310_label_C_term = 0

#: assign_blocks: Only fill gaps in 310-helices that are as large as the
#: following number
assign_blocks_310_gap_tolerance = 3

```

```
# Tables
# -----

#: Render HBond tables with detailed information
hbond_table_detailed = False

#: Sort HBond tables by the hydrogen bond type
hbond_table_sort_type = False

#: Render the Rama table with detailed information, including minor assignments
rama_table_detailed = True
```

Partial Alignment Module

Data Types

Data Readers - Partial Alignment Files

Processing of Magnetic Interactions

SVD

Analysis

Reports

Settings

mollib/pa/settings.py

```
#: The gyromagnetic ratios (in rad T-1 s-1) of common nuclei
gamma = {'H': 267.513E6,
         'C': 67.262E6,
         'N': -27.116E6}

#: Calculate dipolar couplings from bond lengths and gyromagnetic ratios. If
#: this is False, the values in the default_predicted_rdcgs are used. However,
#: if a value is not specified in the default_predicted_rdcgs, the value will
#: still be calculated from bond lengths.
calculate_from_bonds = False

#: The default dipolar couplings calculated.
#: The static dipolar coupling constants listed are for common bonds in
#: proteins. Values are calculated based on average calculated DCCs for 2MJB
#: and scaled to an HN DCC of -11472Hz. This is equal to the
#: dipolar reduced anisotropy.
#: These values are used if ``calculate_from_bonds`` is False.
default_predicted_rdcgs = {'N-H': 10823., #11472., # 1.02 A
                          'NE1-HE1': 10823.,
                          'CA-HA': -22300.,
                          'C-N+1': 1115.,
```

```

        'C-CA': -1880.,
        'CA-CB': -1880.,
    }

#: The default RACS tensor values of various nuclei in proteins.
#: order - the xx/yy/zz component order of the tensor.
#:
#:     1. The first component is colinear with the atom -- ref_atom1
#:        vector
#:
#:     2. The second component is orthogonal to the
#:        atom -- ref_atom1 -- ref_atom2 plane
#:
#:     3. The third component is orthogonal to the second two components.
#:
#:     4. Tensor rotations conducted with the Z-Y-X convention for the
#:        alpha, beta and gamma angles, respectively.
#:
#: See the following reference for tensor conventions and values: Cornilescu
#: et al. JACS 2000, 122, 10143.

default_predicted_racs = {
    # The Carbonyl CSA. This convention places the x-axis (s11) near the,
    # C-N bond and the z-axis (s33) orthogonal to the O-C-N plane.
    'C': {'delta': -86.53 * 1.03, # ppm (Reduced anisotropy)
          'eta': 0.63,
          'alpha': 40., # degrees
          'beta': 0., # degrees
          'gamma': 0., # degrees
          'ref_atom1': 'N+1',
          'ref_atom2': 'O',
          'order': 'xzy',
        },

    # The Nitrogen CSA. This convention places the z-axis (s11) near the H-N
    # bond, and the y-axis (s22) orthogonal to the H-N-CA plane.
    'N': {'delta': 108.53 * 0.993, # ppm (Reduced anisotropy)
          'eta': 0.16,
          'alpha': 0., # degrees
          'beta': -20., # degrees
          'gamma': 0., # degrees
          'ref_atom1': 'H',
          'ref_atom2': 'CA',
          'order': 'zyx',
        },

    # The amide H CSA. This convention places the z-axis (s33) near the H-N
    # bond, and the x-axis (s11) orthogonal to the H-N-C plane.
    'H': {'delta': -5.93, # ppm (Reduced anisotropy)
          'eta': 1.00,
          'alpha': 0., # degrees
          'beta': 0., # degrees
          'gamma': -7., # degrees
          'ref_atom1': 'N',
          'ref_atom2': 'C-1',
          'order': 'xzy',
        },
}

```

```

    },
}

#: Default errors in absolute values
default_error = {'N-H': 0.2, # Hz
                 'CA-HA': 0.5, # Hz
                 'N-C': 0.5, # Hz
                 'CA-C': 0.5, # Hz
                 'NE-HE': 0.2, # Hz
                 'C-C': 0.5, # Hz (bond type)
                 'C-H': 1.0, # Hz (bond type)
                 'C': 1., # ppb
                 'N': 1., # ppb
                 'H': 5., # ppb
                }

#: urls to download mr files
mr_urls = ('https://files.rcsb.org/download/',)

#: Xplor-NIH incorporates C-H methyl RDCs by projecting them on the associated
#: carbon-carbon bond. If the data uses this convention, then set this setting
#: to True. This convention, however, occludes the use of C-C RDCs, like the
#: CA-CB coupling of an ALA, and it is not set by default.
project_methyls = False

#: The order parameter to use for methyl groups
methyl_order_parameter = 1.0

#: Settings to enable individual fixers by default
enable_signfixer = True
enable_outlierfixer = False
enable_nhscalefixer = False

#: The alpha-critical values for the Grubbs test of outliers. These are used to
#: identify warning or bad data points in terms of their deviations with
#: respect to the best-fit SVD data.
alpha_warning = 0.05
alpha_bad = 0.01

```

Statistics Module

Statistics Base Class

Ramachandran Statistics

Hydrogen Bond Statistics

Utilities

General utilities for rendering data to the console

Checks

Markdown

File Utilities

Net Utilities

Interaction Utilities

Tensor Utilities

Rotations

Settings

molib/utis/settings.py

```
#: The default max width (in characters) of markdown tables (MDTable)
#: This is the value used when outputting tables outside of the terminal
default_max_width = 100

#: When writing files, overwrite them if they already exist.
overwrite_files = True

#: Save files fetched and cached files (get_or_fetch) to the local directory
save_fetched_files_locally = False

#: If files are not overwritten, the create new versions up to the following
#: number
max_file_versions = 99
```

Bibliography

[Cornilescu2000] Cornilescu, G. & Bax, A. Measurement of Proton, Nitrogen, and Carbonyl Chemical Shielding Anisotropies in a Protein Dissolved in a Dilute Liquid Crystalline Phase. *J. Am. Chem. Soc.* 122, 10143–10154 (2000).