

PCASSO v1.0

Authors: Sean M. Law
Aaron T. Frank
Charles L. Brooks III*

Affiliation: Department of Chemistry and Biophysics Program,
University of Michigan, Ann Arbor, Michigan, 48109

Website: <http://brooks.chem.lsa.umich.edu/software>

Purpose: PCASSO is a fast and accurate program for assigning protein secondary structure elements using only the C α positions.

Corresponding Author: brookscl@umich.edu

Main Reference: PCASSO: A fast and efficient C α -based method for accurately assigning protein secondary structure elements, **S.M. Law, A.T. Frank, and C.L. Brooks III**, *Submitted*, 2014

Revisions: May 1, 2014 – PCASSO v1.0 released

Source Files and Installation

- Makefile
- lib/
- src/
- tests/

PCASSO is written in C++ and can be compiled by issuing:

```
make
```

at the command line. An executable called “pcasso” will be generated and can be found in the directory called “bin/”

To remove all object files and the executables issue:

```
make clean
```

Command Line Arguments

```
Usage:  pcasso [-options] <PDBfile(s)>
Options: [-verbose]
         [-trj TRAJfile]
         [-skip frames] [-start frame] [-stop frame]
```

The PCASSO program requires at least one PDB file (<PDBfile(s)>) as input along with other optional arguments ([-options]). The order in which the required/optional arguments are provided does not matter.

<PDBfile(s)>

In its most basic form of usage, one or more PDB formatted files maybe be supplied to PCASSO for secondary structure element (SSE) analysis. The PDB files can take on any name and do not need to be of the same molecule. Also, a molecule can contain multiple chains. However, PCASSO will only return the SSE assignments for the first model of a PDB file with multiple models (e.g., NMR structures).

Example 1:

```
pcasso file.1.pdb
```

Example 2:

```
pcasso file.1.pdb file.2.pdb file.3.pdb ... file.N.pdb
```

Example 3:

```
pcasso a.pdb b.pdb c.pdb ... n.pdb
```

Output

In the examples above where only PDB files are used as input, PCASSO will output the name of the PDB file, followed by the time (zero in the case of a PDB file), and a list of single letter SSE assignments (H = helix, E = strand, C = loop/coil). All SSE assignments for each PDB file will appear on a single continuous line.

Example 1 Output:

```
file.1.pdb 0 H H H C C C E E E C C C
```

Example 2 Output:

```
file.1.pdb 0 H H H C C C E E E C C C  
file.2.pdb 0 H H C C C C E E E C C C  
file.3.pdb 0 H H C C C C C E E C C C  
.  
.  
.  
file.N.pdb 0 H H H C C E E E E E E C
```

Example 3 Output:

```
a.pdb 0 H H H H H H H H  
b.pdb 0 C C E E E E E E C C H H H H H H H H H C C C C C C  
C H H H H H H  
c.pdb 0 E E C C H C C E E C C C  
.  
.  
.  
n.pdb 0 C C C C C C C C C C C C C C C C C C C C C C C C C C C C
```

[-options]

The following are options that can be specified at the command line.

-verbose

This option changes the output to include more verbose information about each residue in the form of **CHAIN:RESIDUE.ATOM**.

Example 4:

```
pcasso -verbose file.1.pdb
```

Example 4 Output:

```
file.1.pdb 0 A:GLY1.CA H A:.VAL2.CA H A.ARG3.CA H  
B:THR102.CA C B:SER103.CA C B:ASN104.CA C B:GLN105.CA E  
C:CYS10.CA E C:ALA11.CA E C:TRP12.CA C C:TYR13.CA C  
C:LEU14.CA C
```

-trj TRAJfile

This option allows a CHARMM (binary) trajectory file to be analyzed. Only a single PDB file can be used with this option (or only the first PDB file is used if more than one is provided) and the number of atoms in the PDB must match the number of atoms in the trajectory file. Multiple trajectories (of the same molecule) can also be analyzed with multiple “-trj” arguments. Note that the first and second columns in the output represent the frame count and the simulation time, respectively. “-verbose” can also be used here.

Example 5:

```
pcasso -trj file.1.dcd file.1.pdb
```

Example 5 Output:

```
1 0.1 H H H C C C E E E C C C  
2 0.2 H H C C C C E E E C C C  
3 0.3 H H C C C C C E E C C C  
.  
.  
.  
10 1.0 H H H C C E E E E E E C
```

Example 6:

```
pcasso -trj file.1.dcd -trj file.2.dcd file.1.pdb
```

Example 6 Output:

```
1 0.1 H H H C C C E E E C C C
2 0.2 H H C C C C E E E C C C
3 0.3 H H C C C C C E E C C C
.
.
.
10 1.0 H H H C C E E E E E E C
11 1.1 H H H C C C E E E C C C
12 1.2 H H C C C C E E E C C C
13 1.3 H H C C C C C E E C C C
.
.
.
20 2.0 H H H C C E E E E E E C
```

[-skip frames] [-start frame] [-stop frame]

These options can be used to skip simulation frames and/or control which simulation frames to start/stop on.