XCamShift

XCamShift (XCS) is a new implementation of the camshift chemical shift force field[1] for XPLOR-NIH [2]. The camshift implementation has a number of interesting and useful features

- It is very thoroughly tested against the original camshift implementation in almost [3] and has a large and full test suite which checks that XCS produces results that are identical with camshift in almost.
- XCS uses cython[4] to give a forcefield with native performance written in python
- XCS provides extensions against the standard camshift forcefield including the ability to carry out ensemble calculations, weighting of shift terms and variable well widths to allow for variation in statistical errors in measured chemical shift.
- XCS is modular and individual chemical shift terms can be re weighted during calculations and there is scope for whole components of the force field [e.g hydrogen bonding, aromatic ring currents etc] to be modified or replaced independently.
- XCS provides simple human readable files defining the components of the force field.
- XCS is open source and is free software under the Library Gnu Public License (LGPL)

For information on how to install, compile and use XCS see the sections below.

The Details

Installation

XCamShift can be installed from pre compiled binaries or can be compiled on the users own computer targeting a particular installation of xplor-nih. The

[1] **camshift:** Kohlhoff, K.J. et al. 2009. Fast and accurate predictions of protein NMR chemical shifts from interatomic distances. *Journal of the American Chemical Society*. **131** (39),pp.13894–13895. Robustelli, P. et al. 2010. Using NMR chemical shifts as structural restraints in molecular dynamics simulations of proteins. *Structure*. **18** (8),pp.923–933. Kai Kohlhoff's thesis "Protein Chemical Shifts as Structural Restraints in Molecular Dynamics Simulations, University of Cambridge, England, May, 2008

[2] **xplor-nih:** Schwieters, C.D. et al. 2003. The Xplor-NIH NMR molecular structure determination package. *J. Magn. Reson.* **160** (1),pp.65–73. Schwieters, C.D. et al. 2006. Using Xplor NIH for NMR molecular structure determination. *Progress in Nuclear Magnetic Resonance Spectroscopy*. **48** ,pp.47–62.

[3] **almost:** Fu, B. et al. 2014. ALMOST: An all atom molecular simulation toolkit for protein structure determination. *Journal of computational chemistry*. **35** (14),pp.1101–1105.

[4] **cython** C extensions for python (an optimising static compiler for both the Python programming language and the extended Cython programming language)

precompiled binary python modules includes all the software required to run XCS and you should not require any other software to be installed other than XPLOR-NIH

1. OSX x86_64 (64 bit)
2. OSX i686 (32 bit) - coming
3. Linux x86_64 (64 bit)
4. Linux x86_64 (32 bit) - coming
5. sources: see downloading the source from Github below

before you install you should

1. if md5sum is installed checks the md5sum of the download is correct md5sum XCamShift_xxx.tgz [on OS X you will need md5 -r XCamShift_xxx.tgz instead]. The md5sum is stored in XCamShift_xxx_md5.txt
2. extracts the archive XCamShift_xxx.tgz into a convenient temporary directory using the command tar -zxvf XCamShift_xxx.tgz

the easiest way to install XCamShift is to run the installation command

```
install_dist.sh <dist_dir> <install_dir>
```

which is included with the distribution. When using this command replace <dist_dir> with the path to the directory that contains the xcamshift distribution (this is the directory that contains the install_dist.sh file itself) and install_dir is the directory that contains the particular xplor distribution to install into.

> **NOTE 1.** if you want to use XCamShift without modifying your xplor-nih installation see the section setting paths below

---

> **NOTE 2.** In summary this is what install_dist.sh does (in the following xxx refers to a specific version e.g. v1, <platform> is a computer platform e.g. OSX or Linux and <architecture> is a cpu architecture e.g. x86_64 or i686)
>
> 1. move the .so files in xcamshift_xxx/modules into the folder python/bin.<platform>_<architecture> in you xplor installation; on OSX <platform> will be Darwin_<XX> (where <XX> is the darwin version) and just Linux for linux. <architecture> will be either X86_64 or i686 for 64 and 32 bit computers respectively.

2. Move the directory database/XCamShift which contains the forcefield definition into the database directory of your XPLOR-NIH installation.
3. copy the contents of the camshift_xxx/python directory into the python directory of your xplor distribution
4. test the installation by running the script src/suite.py with the command line pyXplor src/suite.py. You shouldn't see any errors of the form fail or error or abnormal program termination; any of these show a bug is present and should be reported as described in bug reporting below.

Using XCamShift

XCS can only be accessed from the python interface of xplor NIH currently. Here are some snippets showing how to carry out the main operations required to run XCS.

importing shifts

```
csFile=open("test_data/alvin/cs.dat")
cs=csFile.read()
csFile.close()
```

the format of a chemical shift restraint file is detail below under restraints

setting up a calculation

```
from XCamShift import XCamShift
XCamShift.addRestraints(cs)
potList.append(XCamShift)
```

setting up weights

```
XCamShift.setScale(6) # this will change the weights of all terms
```

Format of the xcamshift restraint list

camshift restraint files are typical xplor restraints files and accept acclamation marks (!) as comments that run to the end of lines. The format is

```
<weight-statement>|<class-statement>| <assign-statement>

weight-statement = WEIGht <float>
class-statement = CLASs <four-letter-code>
assign-statement = ASSIgn <selection> <float>|
                   ASSIgn <selection> <float> <float>
```

the weight and <class> statement set weights for all following <assign> statements. statements can take either one and two float values which are in order: the chemical shift and the error of the chemical shift. It is not possible to set individual weights without also setting an error without recourse to the weight statement. The class statement can be used to tag shifts for later manipulation (currently you can just use this to reset the weight of a set of restraints.

An example of an XCamShift restraint file would be

```
class TEST                                     ! this is a comment
assign ( resid 2 and name C ) 177.477          ! restraint with default error
weight 2.0
assign ( resid 3 and name C ) 175.002 0.1      ! error is 0.1 weight 2.0
weight 0.5
assign ( resid 4 and name C ) 176.647 1.0      ! error 1.0 weight 0.5
```

It should be noted that chemical shift restraint files can't currently take sophisticated expressions as provided by the classic xplor syntax.

notes:

1. restraints from the first and last residues in a segment are ignored and a warning issued as these shift cannot be used by camshift as it calculates shifts for the atoms in the central residue of a triple of residues
2. segments with less than 3 residues are ignored and a warning is issued again because camshift as it calculates shifts for the atoms in the central residue of a triple of residues

Restraint objects

Restraint objects can be retrieved from the XCamShift forcefield using the method getRestraints() on the XCamShift potential object. Restraint objects have the following attributes

- obs - the observed shift (can be set)
- calcd - the last calculated shift, if shifts have been calculated
- diff - obs - diff
- err - the err on the observed shift this will be the 1/2 width of the square well where no forces will be applied (can be set)
- weight - the weight of the shift (can be set)
- comment - any end of line comment present when the shift was read in
- name - the name of the atom being restrained (segid resid restype atomname)
- calcd2 - calcd**2
- obs2 - obs**2

Restraint statistics

the XCamShift force field object provides the following methods to retrieve information about the number of restraint and violations

- rms - the rms of diff between obs and calcd (see below for how a threshold is set)
- violations - the number of violations
- numRestraints - the number of restraints

As the range of chemical shifts is not uniformly distributed the value of the threshold [set using the attribute threshold of the forcefield] for restraint violations has to be treated differently and is used as a multiplier of the error. Therefore a restraint is violated when the following equality tests true

```
|diff| > threshold * err + err
```

Listing predicted shifts

The chemical shifts predicted by XCamShift can be printed using the methods print_shifts of the potential. The shifts are printed in the following format

```
SEGID     RESID    RES        HA       CA        HN        N        C        CB
|AGB3|    2        GLN    4.5501  54.2330    8.7379 121.0727 175.1000   29.2632
|AGB3|    3        TYR    5.0569  56.5513    9.5938 123.5669 174.7284   40.6070
|AGB3|    4        LYS    4.8969  54.5661    9.4434 121.8343 174.0538   35.2424
|AGB3|    5        LEU    4.6559  52.7604    9.2490 124.7154 174.3615   43.3438
|AGB3|    6        VAL    4.2273  60.9981    8.9896 124.2064 174.2570   32.7248
|AGB3|    7        ILE    4.2590  59.6785    9.0951 126.6597 174.7850   37.6970
|AGB3|    8        ASN    4.6711  51.7622    9.3382 125.9824 173.3525   38.0188
|AGB3|    9        GLY    4.0863  45.1157    8.6680 110.2235 174.1825         .
|AGB3|    10       LYS    3.9673  58.6702    9.3420 122.0362 176.9272   31.7829
|AGB3|    11       THR    4.2896  63.8501    8.1219 113.1026 174.1789   69.9168
...
|AGB3|    54       VAL    4.4473  59.3198    8.6110 121.8589 173.1164   32.8183
|AGB3|    55       THR    4.8757  60.3260    8.6678 120.6908 173.4372   71.6735
```

note that missing chemical shifts which are structuraly possible e.g. GLY CB are replaced by a . and that segids are wrapped with a || pair to make spaces visible.

Rebuilding XCamShift

You should only need to recompile XCamShift if you want to make changes to the code in XCamShift, re-compile it on a system which isn't supported or use a different compiler. If you only need to recompile with a different compiler or different target version of xplor-nih you won't need cython, you can just recompile the C code produced by cython.

Pre requisites To rebuild the camshift tables (it is unlikely that you will need to do this!) if you are running on an older version of python you may need to build argparse[5] and add it to the PYTHONPATH

Downloading the source from Github source code can be downloaded from the github master branch using the following link: locsmith/xcamshift/archive/master.zip to clone the complete repository us the the following command git clone https://github.com/locsmith/xcamshift.git

Rebuilding the C code

1. Firstly make sure you have a compiler that is compatible with GCC. On linux GCC ~~and ICC~~ have been tested. On OSX I have only used GCC from the homebrew project not Clang.
2. Make sure you have installed the source for your distribution of XPLOR-NIH
3. Download the source code for the version of python you want to target and build it. To test which version of python explore is using run <xplor-nih directory>/bin/pyXplor test.py with test.py containing import sys ; print sys.version. This will give you the version of python, typically the first line of output should be of the form 2.7.5 (default, Mar 9 2014, 22:15:05) and in this case the version is 2.7.5.
4. edit build_all.py and change the variables
5. run build_all.py

Setting Paths XCamShift can be tested without modifying xplor-nih by adding the following to you PYTHONPATH

<XCamShift-directory>/src/cython:<XCamShift-directory>/XCamShift/src

where is the directory that contains the XCamShift distribution

If it is not installed you will also need to install PyYAML[6] and add it's installation directory to the PYTHONPATH variable

[5] **argparse** part of python since version 2.7, for earlier versions of python

[6] **pyyaml:** an implementation of the yaml markup language for python, you may also wish to install libyaml to get a much faster runtime

Testing XCamShift If XCamShift is installed or the paths are correctly set XCamShift can be tested using the following command

```
pyXplor <XCamShift-directory>/src/test/test__XCamShift__print__shifts.py
```

if they are not installed already you will also have to add python paths for the following python packages if you want to run the tests

1. nanotime[7]
2. unittest2[8]

Reporting bugs Bugs happen! XCamShift is well test but new code, please report bugs (or make requests for enhancements or even submit enhancements!) on the XCamShift github site:

The author can also be contacted at g dot s dot thompson at leeds dot ac dot uk

Caveats

1. XCamShift calculates chemical shifts for hydrogen bonds but doesn't calculate any forces in the same manner as the original version. The hydrogen bond shift calculator can be disabled using the following code <xcamshift-potential>.remove__named__sub__potential('HBOND') where <xcamshift-potential> is the xcamshift potential instance.

Tested versions

- **Targeted version of Camshift:** XCamShift is currently tested against version 1.35.0 of camshift which is implemented in almost 1.04.
- **Supported versions of XPLOR-NIH:** XCamShift currently supports version of XPLOR-NIH from 2.31 to 2.35 and the precompiled binaries are compiled against XPLOR-NIH 2.35.Please note that for some versions of xplor-nih before 2.35 you may need to recompile from source as the interface for the ensemble code has changed (this definitely included xplor-nih 2.33).

- **Compiler environment:** XCamShift binaries are compiled with GCC version 4.8.2 on OSX and Cython 0.20.2 (Note Cython is only required if the python code is modified, XCamShift can be re-compiled from the C++ source without Cython being present).
- **Targetted platforms:** XCamShift can be used on OSX and Linux under either a 32Bit or 64Bit intel environment (however 32bit OSX is not tested).

[7] **nanotime:** a nano second resolution timing service for python
[8] **unittest2:** a back port of the unit testing code from python 3+

Citing XCamShift

There is currently no paper for XCamShift (I hope there will be one soon!). Until there is please cite: **XCamShift** G.S.Thompson, Astbury Centre for Structural Molecular Biology, University of Leeds, UK [http://github.com/locsmith/xcamshift]

References