Instructions for the usage of the stand-alone version of ArShift

ArShift (**ar**omatic/**ar**yl chemical **shift** predictor) is written in *R* programming language[1] due to the ease of the web implementation,[2] testing and maintenance of *R* codes, which are crucial in this project, taking into account the possible fast evolution of *ArShift* as more experimental chemical shift data become available. The plans are to make a compiled snapshot implementation of *ArShift* (Fortran03), still keeping the mainstream copy-code in *R* for initial testing of the developments. However, before the Fortran03 version is finalized, the current *ArShift* calculations take long due to the implementation language rather than the algorithm. For a protein with about 25 aromatic rings, the calculation on a single structure takes approximately 4 minutes.

ArShift is platform independent (currently tested on *Linux* and *MacOSX*), as soon as the *R* programming environment is installed. In order to use the stand alone version of *ArShift*, the following steps should be undertaken:

1. Install the *R* version 3.1.1 or later via their download section.

2. Unpack the provided `ArShift_exe.tar.gz` archive file. A folder `ArShift_exe` should be generated with example and *ArShift* programme files inside. This folder will also be the working directory of any *ArShift* calculation.

3. To start a calculation, a protein *pdb* file is needed in `ArShift_exe` directory with added hydrogen atoms. If possible, optimise the hydrogen positions via *Amber03* force field.[3] Internally, all the *Lys* residues are treated as protonated, all the *His* residues are treated as ʃ-protonated (provide ʃ-protonated files for a better precision), and all the *Cys* residues are accounted for in their free state. However, the naming of the amino acid residues should comply the usual *pdb* convention (`HIS`, `CYS`, `LYS` etc.) in the input files, hence, please, avoid using force field specific notations such as `LYP`, `CYN`, `HID`, `HIE`, `HISE`, `HISD` etc. *ArShift* attempts to recognise and correctly identify different atom naming convention inside *pdb* files (`1HB2`, `HB21` etc.). If average chemical shifts are needed to be evaluated for a conformational ensemble, a single *pdb* file with multiple conformers can be supplied, provided that the transition between conformers include the conventional *TER* and *MODEL* lines as in standard *pdb* files.

4. Optionally, if comparison with the experimental data is required, an experimental data file can be supplied to *ArShift* by putting it in the `ArShift_exe` directory. The lines in the file should have the following order in a free format:

```
# RESIDUE SEQUENCE PDB_NUCLEUS_TYPE CHEMICAL_SHIFT CHAIN
PHE 4 HD 7.036 A
```

Any line that begins with "`#`" will be considered as a comment and discarded. Please supply only a single chemical shift entry for both `HD1` and `HD2` or `HE1` and `HE2`. If the residue sequence numbers in the experimental data file are shifted from the numbers in the *pdb* file, the correct value of the `seqshift` command in the `command.cmd` file should be specified (see below). The chain id should be provided for multi-chain systems and can only be skipped if you try to calculate a single chain, the *pdb* file of which does not have the chain id section populated.

5. The `command.cmd` file should be modified accordingly, with the command keywords holding the following meanings:

`title` - anything to help you identify the calculation from the output. Do not use long specifications. If nothing is specified by either setting it to "" or commenting that out via "`#`", then "`Not specified`" will be printed as its internal default.

`pdbname` - the name of the *pdb* file (with file extension).

`pdbtype` - this specifies the type of the parser to be used in reading in the *pdb* file. The default and safest one is the "`complex`", which should recognise the majority of *pdb* file conventions. The other options ("`medium`"and "`simple`") are for a more standard *pdb* file and will work a little bit quicker which can save a minute in case a calculations are done on a very large set of conformers. For *Almost*[4] processed files, the option "`almost`" should be specified.

`seqshift` - if the residue sequence in the supplied *pdb* file is shifted from the sequence in the experimental data file, or if one wants to alter the sequence numbers in the *ArShift* output, then the `seqshift` keyword can be used that accepts both negative and positive numbers. The default value is `0`.

`examine` - a keyword specifying the order number of the conformer to be analysed. If `0` (default), all the conformers (or the single one if there are no multiple structures) within the supplied *pdb* file will be analysed and only the averaged chemical shift values will be printed. Any other integer positive number will imply that only the corresponding (1st, 2nd, etc.) conformer in the *pdb* file will be analysed.

`experdata` - the name of the experimental chemical shift data file (with file extension). If no experimental data exist, `NULL` (internal default) or "" should be provided.

`rereference` - a boolean keyword (`TRUE` or `FALSE`). If `TRUE`, the predicted chemical shifts will be re-referenced to provide a better match to the experimental data via an offset determined by least squares fitting, given that the experimental data exist with at least 10 chemical shift assignments.

`outputname` - the name of the output file (with file extension). The internal default is "`out.txt`".

`outorder` - the order of the results ("`bytype`" - default selection to sort the output by residue type, "`byseq`" - by residue sequence number, "`byshift`" - by the predicted chemical shift value) in the output file.

`bias` - a positive integer parameter controlling the colour pallet for writing a chimaera command file that can colour the structure in accordance to the structural quality as validated via the supplied chemical shifts. Higher values give a more widely spaced colour in the high end. The internal default, `2`, is highly recommended.

6. The calculation can finally be launched from either the operating system command line, or *R* terminal. In order to launch *ArShift* from *R* terminal, start *R* (usually by typing `R`) and type `source("ArShift.R")`, where `ArShift.R` is the interfacing script to the *ArShift* engine. From the operating system command line, *ArShift* can either be run as a batch job (`R CMD BATCH [options_optional] ArShift.R [logfile]`) or via direct incorporation into other programmes (including shell scripts in Linux). A straight-forward way to make *R* be supported by *Linux bash* shell is the installation of little *r*, *littler* (for Ubuntu, type "`sudo apt-get install littler`"). Then any *R* command or script can be launched directly from the shell. To execute *ArShift*, simply type: `r -e 'source("ArShift.R")'` .

Cite ArShift:

Sahakyan A. B., Vranken W. F., Cavalli A., Vendruscolo M. “Using Side-Chain Aromatic Proton Chemical Shifts for a Quantitative Analysis of Protein Structures”, *Angew. Chem. Int. Ed.*, 50, 9620-9623, **2011**.

Questions:

Bug reports and questions to **Aleksandr Sahakyan** via aleksahak *[*at*] cantab.net* or *as952 [*at*] cam.ac.uk*, attaching all the related files and error messages produced by *ArShift*. Other enquires to Prof. **Michele Vendruscolo** via mv245 *[*at*] cam.ac.uk.*

Note:

If you experience an error similar to the following:

```
Error in record.variants$residue.seq : $ operator is
invalid for atomic vectorsCalls: source ... eval.with.vis ->
convert.topology -> sapply -> lapply -> FUN*
```

it means that the supplied *pdb* file uses an atom naming convention, that is not recognised by the predictor. Please try to remediate the *pdb* file or send the file-example (see above), so that the new convention is added in Ar*Shift* library.

References:

1. R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, *http://www.R-project.org*, **2015**.
2. Newton, R.; Wernisch, L. Rwui: A Web Application to Create User Friendly Web Interfaces for R Scripts, *http://rwui.cryst.bbk.ac.uk*, **2010**.
3. Duan, Y.; Wu, C.; Chowdhury, S.; Lee, M. C.; Xiong, G.; Zhang, W.; Yang, R.; Cieplak, P.; Luo, R.; Lee, T.; Caldwell, J.; Wang, J.; Kollman, P. *J. Comput. Chem.* **2003**, 24, 1999–2012.
4. Fu B., Sahakyan A. B., Camilloni C., Tartaglia G. G., Paci E., Caflisch A., Vendruscolo M., Cavalli A., *J. Comput. Chem.*, 35, 1101-1105, **2014**.